

IQI 04, Seminar 10/11

Produced with pdflatex and xfig

- Search problems.
- Unstructured search.
- Grover's algorithm.
- Quantum counting.

E. "Manny" Knill: knill@boulder.nist.gov

Examples of Search Problems

- BITSEARCH.

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Examples of Search Problems

- BITSEARCH.

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$.

Examples of Search Problems

- BITSEARCH.

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Examples of Search Problems

- BITSEARCH.

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

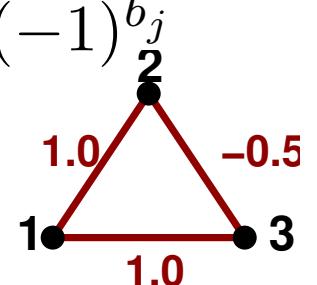
- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$

that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

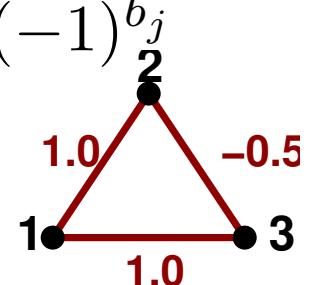
Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1b_2\dots b_n$

that minimizes the energy $\sum_{i,j} J_{i,j}(-1)^{b_i}(-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$.

Solution: 100 or 011.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

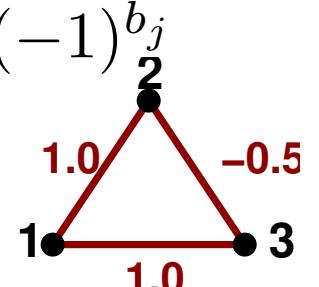
Problem: Find a configuration $b = b_1 b_2 \dots b_n$

that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$.

Solution: 100 or 011.

Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

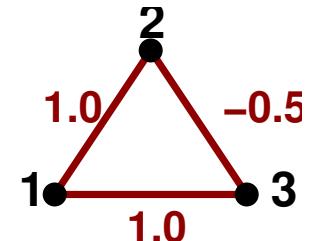
- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$. Solution: 100 or 011.

Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

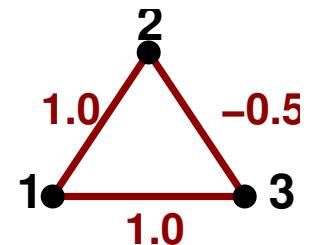
Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$. Solution: 100 or 011.

Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.

- **OPTREVNET.**

Input: n -gate $c^2\text{not}$ network \mathcal{N} on k bits.

Problem: Find the smallest $c^2\text{not}$ network that implements
the same function as \mathcal{N} .



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$. Solution: 100 or 011.

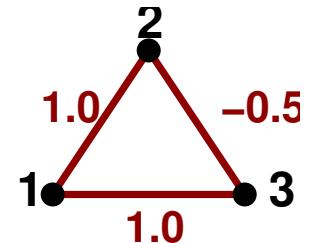
Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.

- **OPTREVNET.**

Input: n -gate $c^2\text{not}$ network \mathcal{N} on k bits.

Problem: Find the smallest $c^2\text{not}$ network that implements
the same function as \mathcal{N} .

Input complexity: $\text{bitlength}(\mathcal{N})$.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$. Solution: 100 or 011.

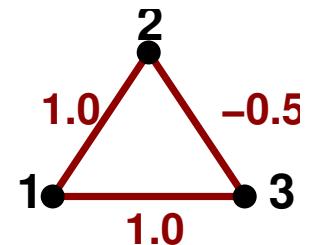
Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.

- **OPTREVNET.**

Input: n -gate $c^2\text{not}$ network \mathcal{N} on k bits.

Problem: Find the smallest $c^2\text{not}$ network that implements
the same function as \mathcal{N} .

Input complexity: $\text{bitlength}(\mathcal{N})$.



Examples of Search Problems

- **BITSEARCH.**

Input: Bitstring b .

Problem: Find the position of a 1 in b if there is a 1 in b .

Example: $b = 0010100$. Solution: 3 or 5.

Input complexity: $|b| = \text{bitlength}(b)$.

- **MINISING.**

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems.

Problem: Find a configuration $b = b_1 b_2 \dots b_n$
that minimizes the energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j}$

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$. Solution: 100 or 011.

Input complexity: $\sum_{i,j} \text{bitlength}(J_{i,j})$.

- **OPTREVNET.**

Input: n -gate $c^2\text{not}$ network \mathcal{N} on k bits.

Problem: Find the smallest $c^2\text{not}$ network that implements
the same function as \mathcal{N} .

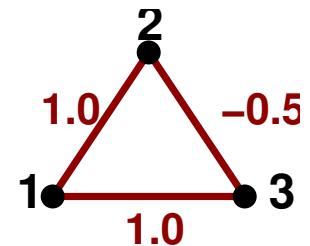
Input complexity: $\text{bitlength}(\mathcal{N})$.

- **CHECKERSMOVE.**

Input: A “checkers” position on an $n \times n$ gameboard.

Problem: Find a winning move for “black”, if such a move exists.

Input complexity: n^2 .



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?.

Example: $b = 0010100$.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?.

Example: $b = 0010100$. Answer: “yes”.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?.

Example: $b = 0010100$. Answer: "yes".



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy

Problem: Is there a configuration $b = b_1 b_2 \dots b_n$
with energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j} < E$?



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

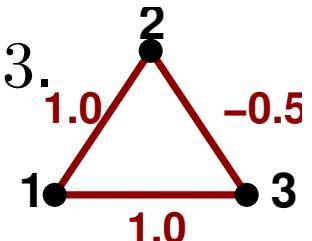
- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy

Problem: Is there a configuration $b = b_1b_2 \dots b_n$

with energy $\sum_{i,j} J_{i,j}(-1)^{b_i}(-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

- BELOWISING.

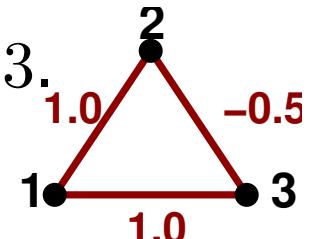
Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy

Problem: Is there a configuration $b = b_1b_2 \dots b_n$

with energy $\sum_{i,j} J_{i,j}(-1)^{b_i}(-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$.

Answer: No.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

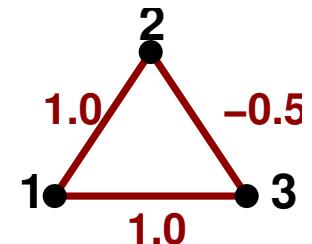
- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy E .

Problem: Is there a configuration $b = b_1 b_2 \dots b_n$

with energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$. Answer: No.



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy E .

Problem: Is there a configuration $b = b_1 b_2 \dots b_n$

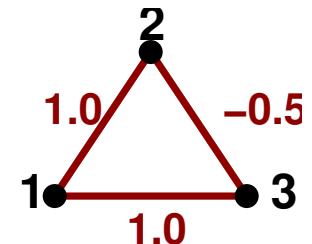
with energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$. Answer: No.

- BETTERREVNET.

Input: n -gate $c^2\text{not}$ network \mathcal{N} on k bits.

Problem: Is there a smaller network implementing the same function as \mathcal{N} ?



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy E .

Problem: Is there a configuration $b = b_1 b_2 \dots b_n$

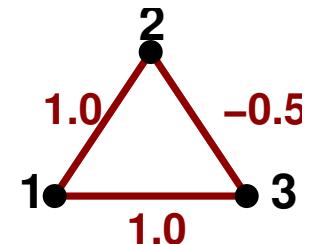
with energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$. Answer: No.

- BETTERREVNET.

Input: n -gate c^2 not network \mathcal{N} on k bits.

Problem: Is there a smaller network implementing the same function as \mathcal{N} ?



Examples of Decision Problems

- EXISTSBIT.

Input: Bitstring b .

Problem: Does b have a 1?

Example: $b = 0010100$. Answer: "yes".

- BELOWISING.

Input: Coupling network $\{J_{i,j}\}$ for n two-level systems, energy E .

Problem: Is there a configuration $b = b_1 b_2 \dots b_n$

with energy $\sum_{i,j} J_{i,j} (-1)^{b_i} (-1)^{b_j} < E$?

Example: $n = 3$, $J_{1,2} = J_{1,3} = 1$, $J_{2,3} = -0.5$, $E = -3$. Answer: No.

- BETTERREVNET.

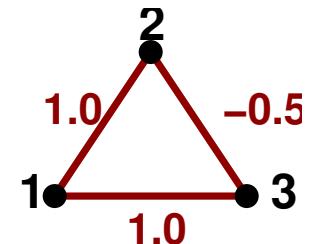
Input: n -gate c^2 not network \mathcal{N} on k bits.

Problem: Is there a smaller network implementing the same function as \mathcal{N} ?

- WINCHECKERS.

Input: A "checkers" position on an $n \times n$ gameboard.

Problem: Does "black" have a winning strategy?



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. \ y \text{ encodes a configuration with energy } \leq E.]$



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. y \text{ encodes a configuration with energy } \leq E.]$
- $R(x, y, \dots)$ is *polynomial time* (is in **P**) if for some k there exists a deterministic classical algorithm that computes $R(x, y, \dots)$ in time $\leq \text{bitlength}(x, y, \dots)^k$.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. y \text{ encodes a configuration with energy } \leq E.]$
- $R(x, y, \dots)$ is *polynomial time* (is in **P**) if for some k there exists a deterministic classical algorithm that computes $R(x, y, \dots)$ in time $\leq \text{bitlength}(x, y, \dots)^k$.

Examples: $\text{EXISTSBIT}(x)$ and $\text{BELOWISING}(x, y)$ are in **P**.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. y \text{ encodes a configuration with energy } \leq E.]$
- $R(x, y, \dots)$ is *polynomial time* (is in **P**) if for some k there exists a deterministic classical algorithm that computes $R(x, y, \dots)$ in time $\leq \text{bitlength}(x, y, \dots)^k$.
- **Examples:** $\text{EXISTSBIT}(x)$ and $\text{BELOWISING}(x, y)$ are in **P**.
- $R(x)$ is *non-deterministic polynomial time* (is in **NP**) if for some k and $Q(x, y)$ in **P**, $R(x) = \exists y (|y| \leq |x|^k \text{ and } Q(x, y))$.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. y \text{ encodes a configuration with energy } \leq E.]$

- $R(x, y, \dots)$ is *polynomial time* (is in **P**) if for some k there exists a deterministic classical algorithm that computes $R(x, y, \dots)$ in time $\leq \text{bitlength}(x, y, \dots)^k$.

Examples: $\text{EXISTSBIT}(x)$ and $\text{BELOWISING}(x, y)$ are in **P**.

- $R(x)$ is *non-deterministic polynomial time* (is in **NP**) if for some k and $Q(x, y)$ in **P**, $R(x) = \exists y (|y| \leq |x|^k \text{ and } Q(x, y))$.

Examples:

- $\text{BELOWISING}(x) = \exists y \text{BELOWISING}(x, y)$.



Decision Problems in P, NP

- A *Decision problem* or *language* is a relation $R(x, y, \dots)$ of one or more strings.

Examples:

- $\text{EXISTSBIT}(x) = [x \text{ has a } 1]$.
- $\text{BELOWISING}(x, y) = [x \text{ encodes } \{J_{i,j}\}, E. y \text{ encodes a configuration with energy } \leq E.]$

- $R(x, y, \dots)$ is *polynomial time* (is in **P**) if for some k there exists a deterministic classical algorithm that computes $R(x, y, \dots)$ in time $\leq \text{bitlength}(x, y, \dots)^k$.

Examples: $\text{EXISTSBIT}(x)$ and $\text{BELOWISING}(x, y)$ are in **P**.

- $R(x)$ is *non-deterministic polynomial time* (is in **NP**) if for some k and $Q(x, y)$ in **P**, $R(x) = \exists y (|y| \leq |x|^k \text{ and } Q(x, y))$.

Examples:

- $\text{BELOWISING}(x) = \exists y \text{BELOWISING}(x, y)$.
- $\text{NONPRIME}(x) = \exists y [1 < y < x \text{ and } x = z * y]$.



NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .

Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.



NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .
Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.
- S is **NP easy** if for some Q in **NP**, S is in \mathbf{P}^Q .

NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .
Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.
- S is **NP easy** if for some Q in **NP**, S is in \mathbf{P}^Q .
 - Note: $[R \text{ is NP complete}] \not\equiv [R \text{ is NP hard and NP easy}]$.

NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .
Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.
- S is **NP easy** if for some Q in **NP**, S is in \mathbf{P}^Q .
 - Note: [R is **NP complete**] $\not\Rightarrow$ [R is **NP hard** and **NP easy**].
- **MINISING** and **BELOWISING** are **NP hard** and **NP easy**.
... **BELOWISING** is **NP complete**.

NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .
Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.
- S is **NP easy** if for some Q in **NP**, S is in \mathbf{P}^Q .
 - Note: [R is **NP complete**] $\not\Rightarrow$ [R is **NP hard** and **NP easy**].
- **MINISING** and **BELOWISING** are **NP hard** and **NP easy**.
... **BELOWISING** is **NP complete**.
- **BETTERREVNET** may not be **NP easy**.

NP Completeness and Hardness

- S is **NP hard** if for every Q in **NP**, Q is in \mathbf{P}^S .
Def.: \mathbf{P}^S means “polynomial time given an oracle for S ”.
- S is **NP easy** if for some Q in **NP**, S is in \mathbf{P}^Q .
 - Note: [R is **NP complete**] $\not\Rightarrow$ [R is **NP hard** and **NP easy**].
- **MINISING** and **BELOWISING** are **NP hard** and **NP easy**.
... **BELOWISING** is **NP complete**.
- **BETTERREVNET** may not be **NP easy**.
- **WINCHECKERS** is “**PSPACE** complete”, hence not expected to be **NP easy**.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.

- Any problem $\exists y (|y| \leq |x|^k \text{ and } R(x, y))$ in **NP** can be solved for a given x by using \mathcal{A} with $m = |x|^k$, $\text{BB}_x(y) = R(x, y)$.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.
- Any problem $\exists y (|y| \leq |x|^k \text{ and } R(x, y))$ in **NP** can be solved for a given x by using \mathcal{A} with $m = |x|^k$, $\text{BB}_x(y) = R(x, y)$.

Unstructured: Does not use prior knowledge about the internals of BB.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.

- Any problem $\exists y \left(|y| \leq |x|^k \text{ and } R(x, y) \right)$ in **NP** can be solved for a given x by using \mathcal{A} with $m = |x|^k$, $\text{BB}_x(y) = R(x, y)$.

Unstructured: Does not use prior knowledge about the internals of BB.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.

- Any problem $\exists y \left(|y| \leq |x|^k \text{ and } R(x, y) \right)$ in NP can be solved for a given x by using \mathcal{A} with $m = |x|^k$, $\text{BB}_x(y) = R(x, y)$.

Unstructured: Does not use prior knowledge about the internals of BB.

- q BBSEARCH.

Given: “Black Box” operator $q\text{BB}|x\rangle|a\rangle = |x\rangle|a + \text{BB}(x)\rangle$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.



Unstructured Search

- BBSEARCH.

$$x \in \{s \mid |s| \leq m\}$$

Given: “Black Box” function $\text{BB}(x) \in \{0, 1\}$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

- Examples:

- To solve BELOWISING using an algorithm $\mathcal{A}(m, \text{BB})$ for BBSEARCH, let $\text{BB}_{\{J_{i,j}\}, E}(C) = 1$ if and only if C is a configuration with energy below E . Use $\mathcal{A}(n, \text{BB}_{\{J_{i,j}\}, E})$.

- Any problem $\exists y \left(|y| \leq |x|^k \text{ and } R(x, y) \right)$ in NP can be solved for a given x by using \mathcal{A} with $m = |x|^k$, $\text{BB}_x(y) = R(x, y)$.

Unstructured: Does not use prior knowledge about the internals of BB.

- q BBSEARCH.

Given: “Black Box” operator $q\text{BB}|x\rangle|a\rangle = |x\rangle|a + \text{BB}(x)\rangle$.

Problem: Find an x such that $\text{BB}(x) = 1$ if such an x exists.

... x and a are restricted to $x \in \{0, \dots, N\}$, $a \in \{0, 1\}$.



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that BB(x) = 1 or “no” if no such x exists.

for $x = 0$ **to** $x = 2^n - 1$

if BB(x) = 1 **then return** x

end

return “no”



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $BB(x) = 1$ or “no” if no such x exists.

```
for  $x = 0$  to  $x = 2^n - 1$ 
    if BB( $x$ ) = 1 then return  $x$ 
end
return “no”
```



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $BB(x) = 1$ or “no” if no such x exists.

```
for  $x = 0$  to  $x = 2^n - 1$ 
    if BB( $x$ ) = 1 then return  $x$ 
end
return “no”
```

- Worst-case number of queries is 2^n .



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $\text{BB}(x) = 1$ or “no” if no such x exists.

```
for  $x = 0$  to  $x = 2^n - 1$ 
    if  $\text{BB}(x) = 1$  then return  $x$ 
end
return “no”
```

- Worst-case number of queries is 2^n .

- Probabilistic search.

PROBSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $\text{BB}(x) = 1$ or “no” if no such x exists.

```
repeat
     $x \leftarrow \text{RAND}([2^n] \setminus X); X \leftarrow X \cup \{x\}$ 
```

```
until  $\text{BB}(x) = 1$  or  $X = [2^n]$ 
```

```
if  $\text{BB}(x) = 1$  then return  $x$  else return “no”
```



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $\text{BB}(x) = 1$ or “no” if no such x exists.

```
for  $x = 0$  to  $x = 2^n - 1$ 
    if  $\text{BB}(x) = 1$  then return  $x$ 
end
return “no”
```

- Worst-case number of queries is 2^n .

- Probabilistic search.

PROBSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $\text{BB}(x) = 1$ or “no” if no such x exists.

```
repeat
     $x \leftarrow \text{RAND}([2^n] \setminus X); X \leftarrow X \cup \{x\}$ 
until  $\text{BB}(x) = 1$  or  $X = [2^n]$ 
if  $\text{BB}(x) = 1$  then return  $x$  else return “no”
```



Classical Algorithms for Unstructured Search

- Deterministic search.

DETSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $BB(x) = 1$ or “no” if no such x exists.

```
for  $x = 0$  to  $x = 2^n - 1$ 
    if BB( $x$ ) = 1 then return  $x$ 
end
return “no”
```

- Worst-case number of queries is 2^n .

- Probabilistic search.

PROBSEARCH(BB)

Input: BB : $\{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

Output: x such that $BB(x) = 1$ or “no” if no such x exists.

```
repeat
     $x \leftarrow \text{RAND}([2^n] \setminus X); X \leftarrow X \cup \{x\}$ 
until BB( $x$ ) = 1 or  $X = [2^n]$ 
if BB( $x$ ) = 1 then return  $x$  else return “no”
```

- If a solution exists, expected number of queries $\leq (2^n + 1)/2$.

egin



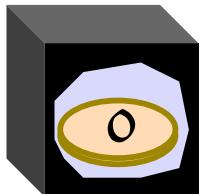
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



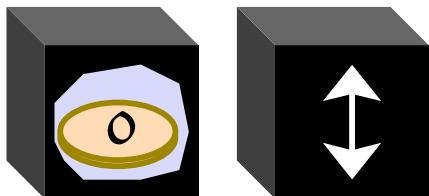
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



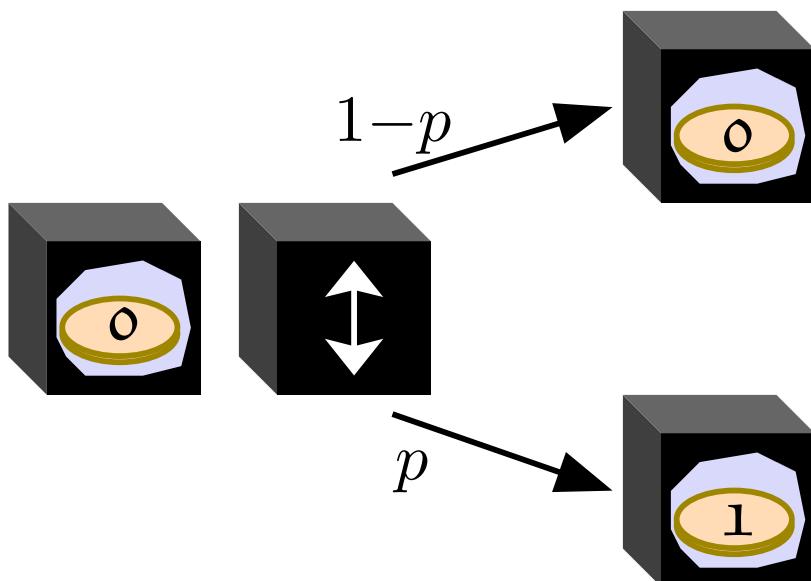
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



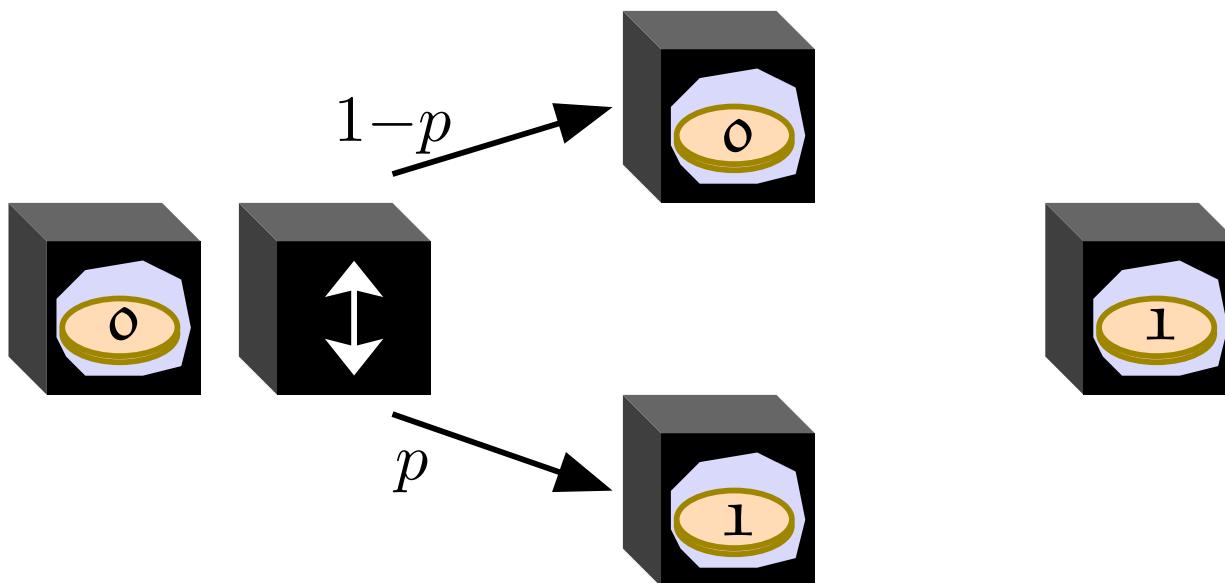
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



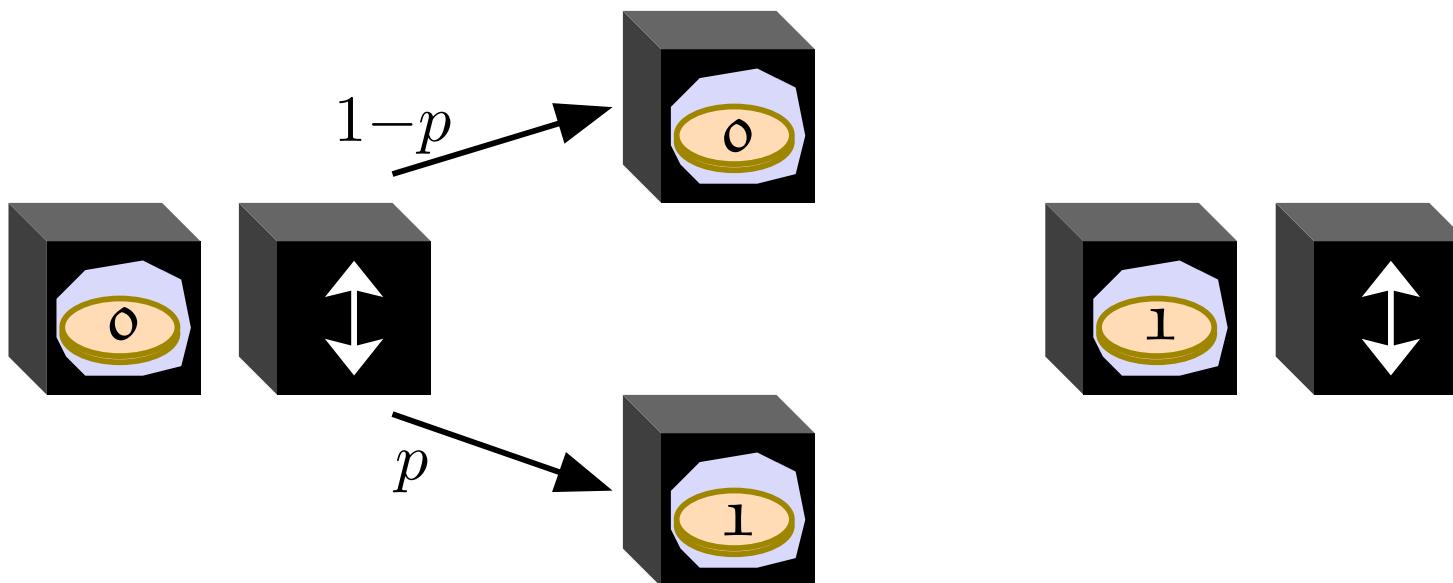
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



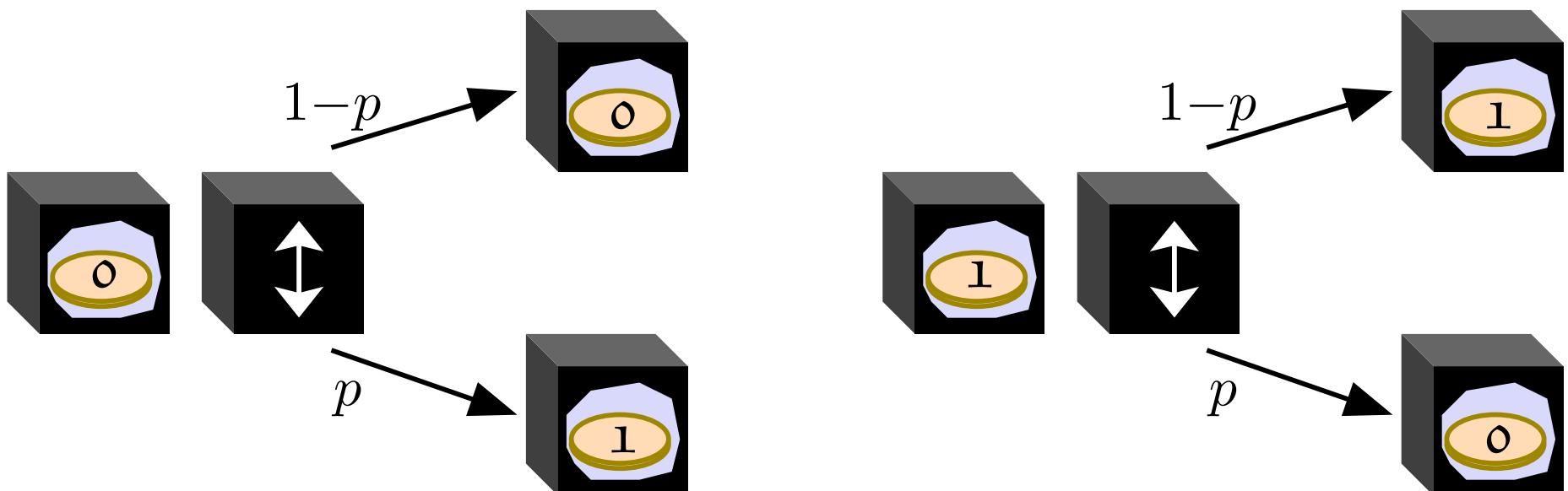
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



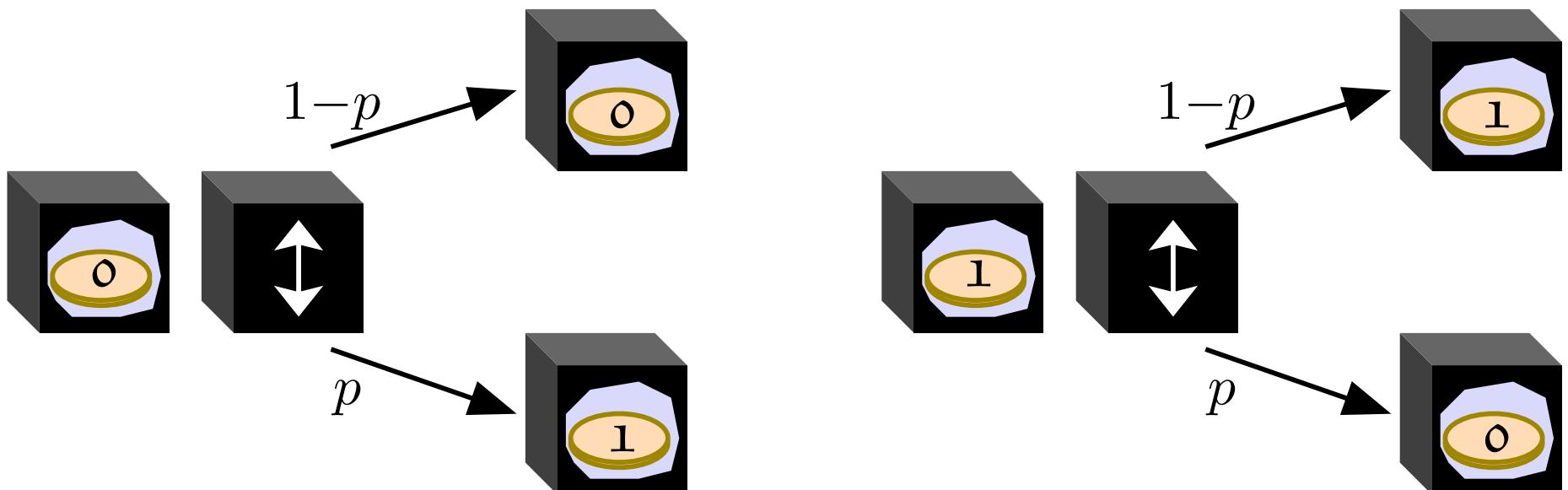
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
- Problem: Determine p .



- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.

Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.

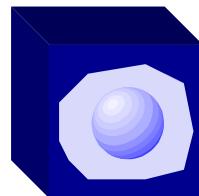


Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2\arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .

Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2\arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .

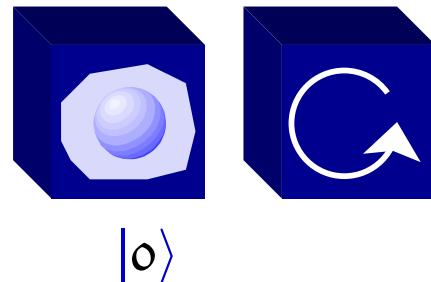


$|0\rangle$



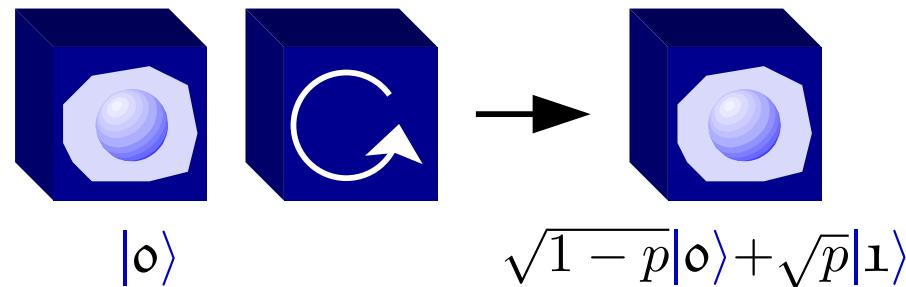
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $Y_{2\arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .



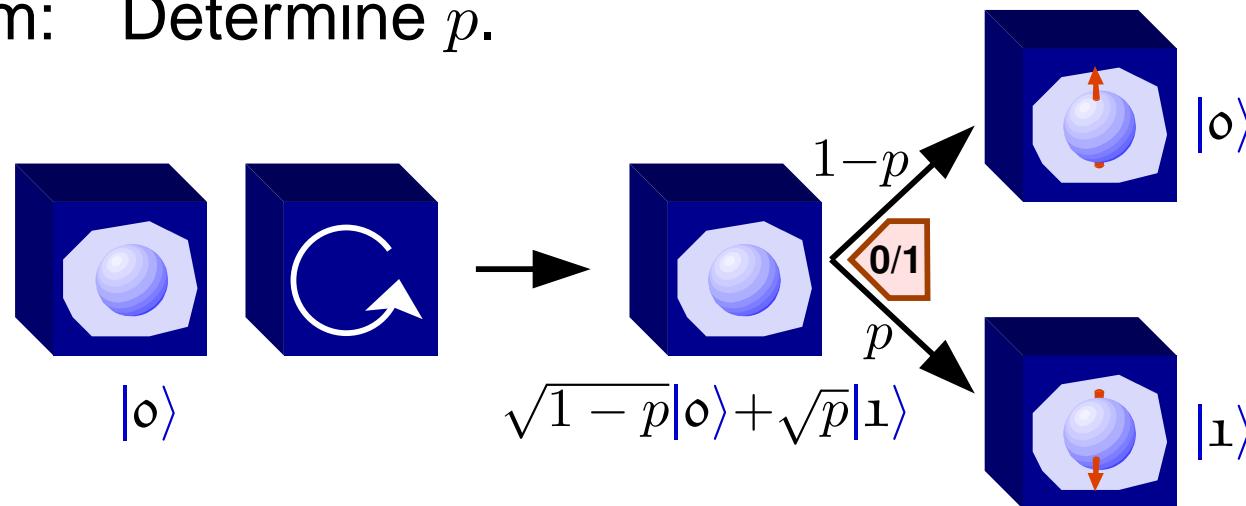
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2\arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .



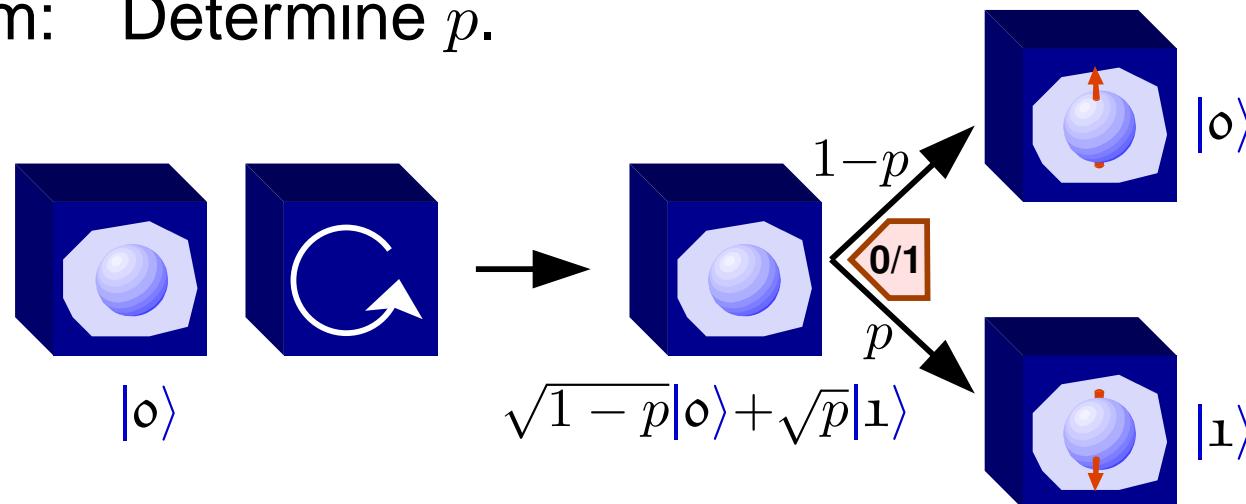
Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $Y_{2\arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .



Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2 \arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .



- Turn n times: $|0\rangle \rightarrow \cos(n \arcsin(\sqrt{p}))|0\rangle + \sin(n \arcsin(\sqrt{p}))|1\rangle$.
Prob. of detecting $|1\rangle$ is $\simeq n^2 p$ for $n^2 \ll 1/p$.

Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2 \arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .
- Turn n times: $|\text{o}\rangle \rightarrow \cos(n \arcsin(\sqrt{p}))|\text{o}\rangle + \sin(n \arcsin(\sqrt{p}))|\text{1}\rangle$.
Prob. of detecting $|\text{1}\rangle$ is $\simeq n^2 p$ for $n^2 \ll 1/p$.



Probabilities versus Quantum Amplitudes

- Given: Box with bit.
A shake flips the bit with probability $p = 0$ or $p = \epsilon$.
Problem: Determine p .
- Shake n times: Prob. of ≥ 1 flip is $1 - (1 - p)^n \simeq np$ for $n \ll 1/p$.
- Given: Box with qubit.
A turn applies $\mathbf{Y}_{2 \arcsin(\sqrt{p})}$, $p = 0$ or $p = \epsilon$
Problem: Determine p .
- Turn n times: $|\text{o}\rangle \rightarrow \cos(n \arcsin(\sqrt{p}))|\text{o}\rangle + \sin(n \arcsin(\sqrt{p}))|\text{1}\rangle$.
Prob. of detecting $|\text{1}\rangle$ is $\simeq n^2 p$ for $n^2 \ll 1/p$.
- Complexity. Probabilistically: $\Omega(1/p)$.
Quantumly: $\Omega(1/\sqrt{p})$.



Grover's Algorithm: States

- Given: $q\text{BB}$ such that $x \in \{1, \dots, N\}, b \in \{0, 1\}$
 $q\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x=u]\rangle_T$ with u unknown.
- Problem: Determine u .

Grover's Algorithm: States

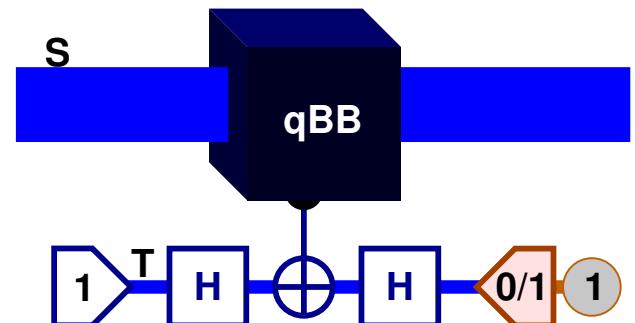
- Given: $q\text{BB}$ such that $x \in \{1, \dots, N\}, b \in \{0, 1\}$

$$q\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+ [x=u]\rangle_T \text{ with } u \text{ unknown.}$$

Problem: Determine u .

- Use phase-kickback to construct

$$z\text{BB}|x\rangle_S = (-1)^{[x=u]}|x\rangle_S.$$



Grover's Algorithm: States

- Given: $q\text{BB}$ such that $x \in \{1, \dots, N\}, b \in \{0, 1\}$

$$q\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+ [x=u]\rangle_T \text{ with } u \text{ unknown.}$$

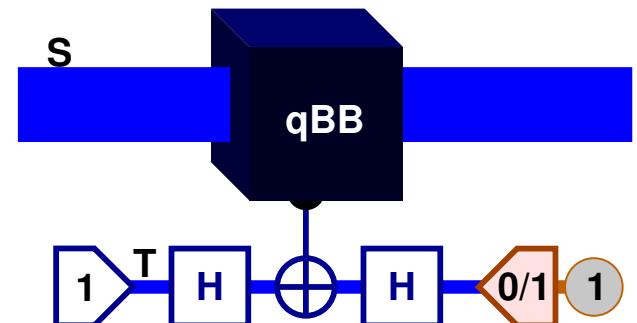
Problem: Determine u .

- Use phase-kickback to construct

$$z\text{BB}|x\rangle_S = (-1)^{[x=u]}|x\rangle_S.$$

- Idea:

Apply $z\text{BB}$ in quantum parallel, amplify the amplitude of $|u\rangle_S$.



Grover's Algorithm: States

- Given: $q\text{BB}$ such that $x \in \{1, \dots, N\}, b \in \{0, 1\}$

$$q\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+ [x=u]\rangle_T \text{ with } u \text{ unknown.}$$

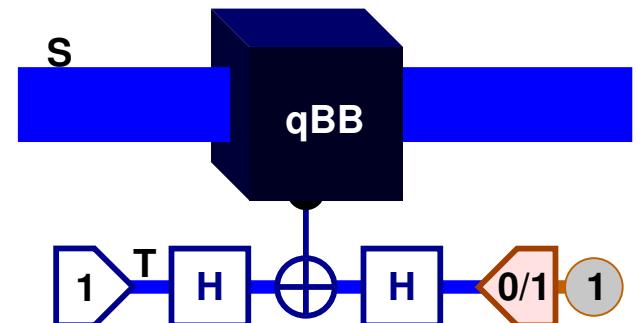
Problem: Determine u .

- Use phase-kickback to construct

$$z\text{BB}|x\rangle_S = (-1)^{[x=u]}|x\rangle_S.$$

- Idea:

Apply $z\text{BB}$ in quantum parallel, amplify the amplitude of $|u\rangle_S$.



- How can one “rotate” from $\frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$?



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.

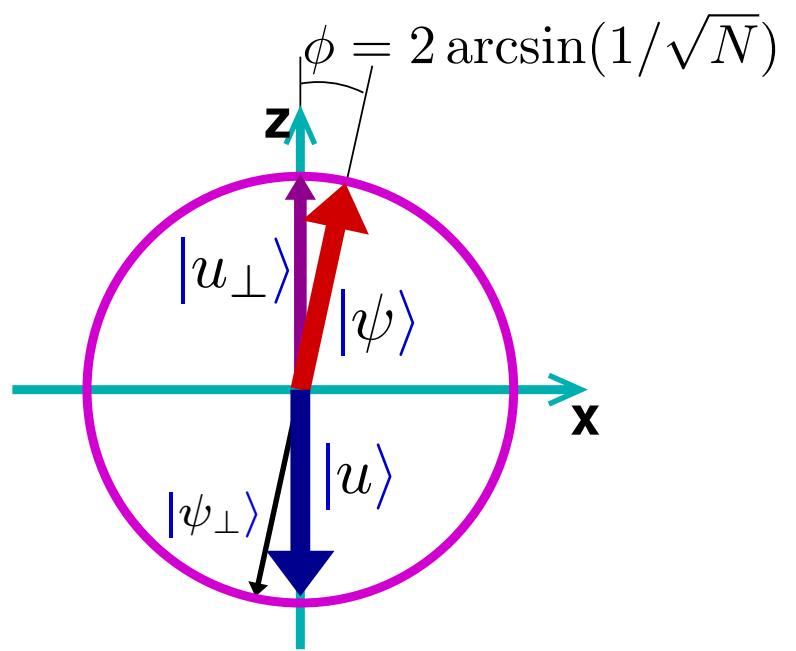


Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:



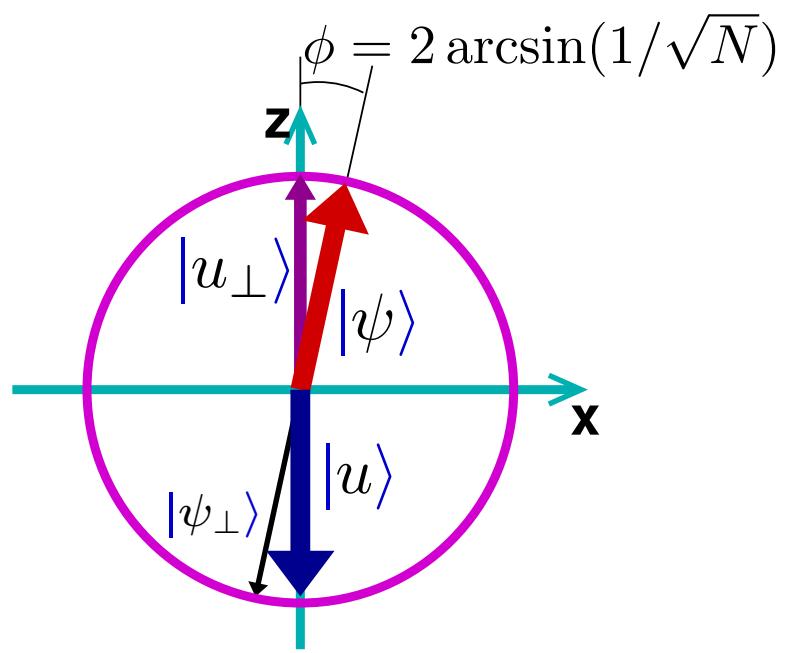
Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.



Grover's Algorithm: Rotations

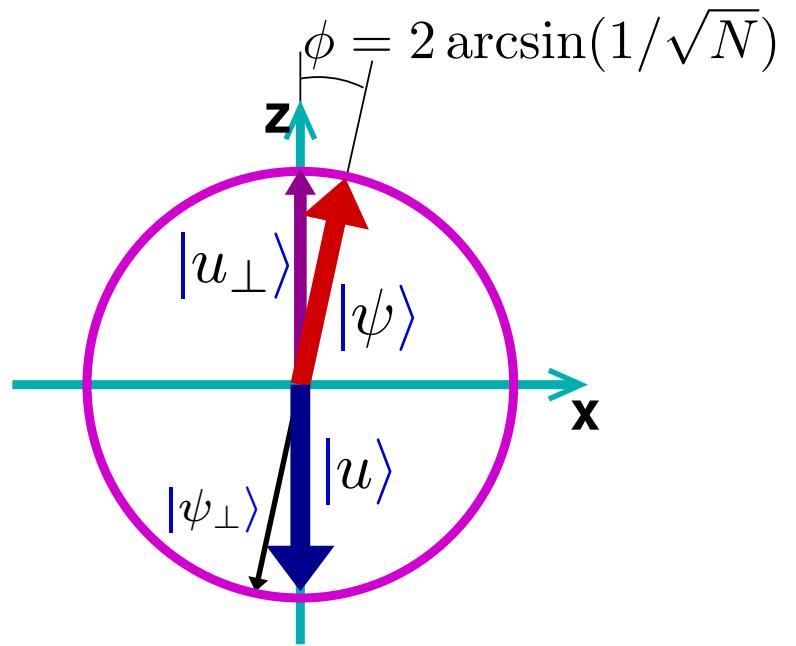
- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle),$$



Grover's Algorithm: Rotations

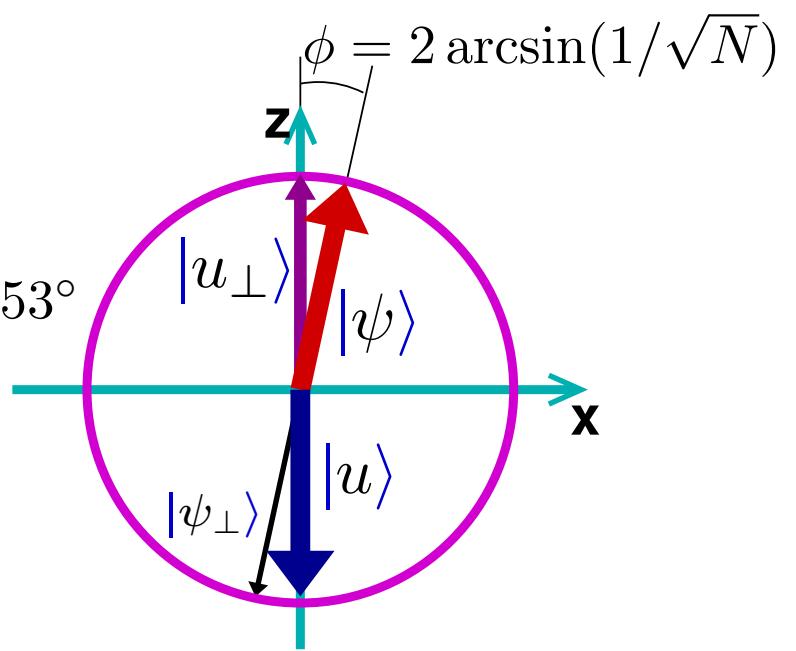
- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$



Grover's Algorithm: Rotations

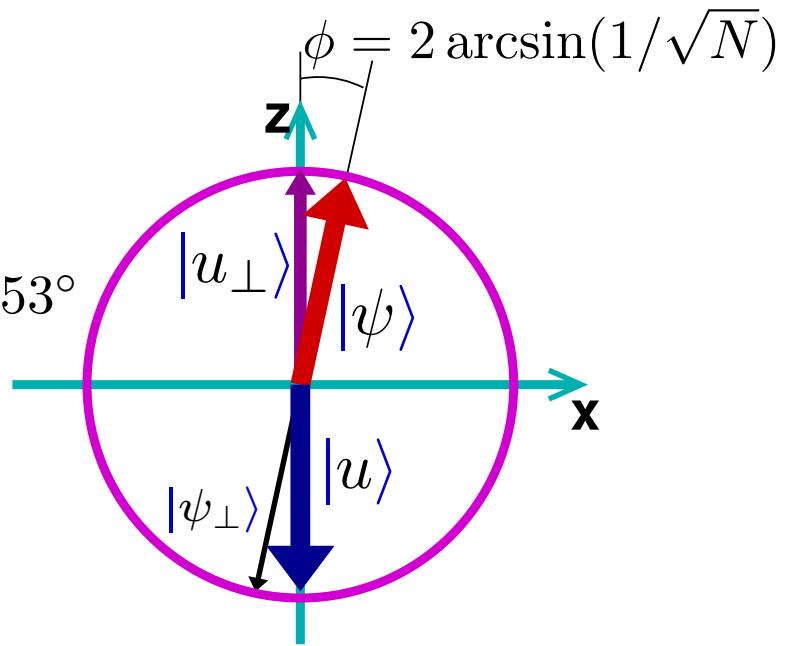
- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$



Grover's Algorithm: Rotations

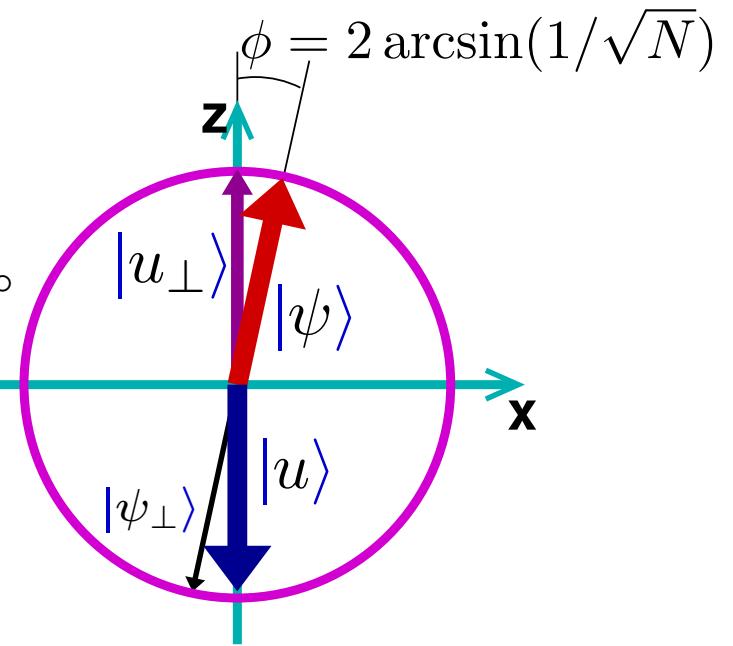
- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

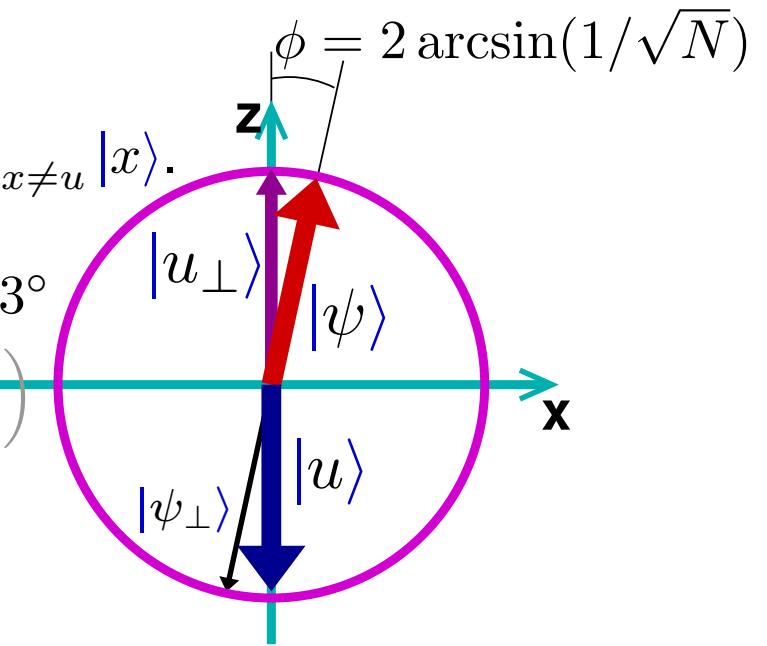
Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

$$|u_\perp\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq u} |x\rangle.$$



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

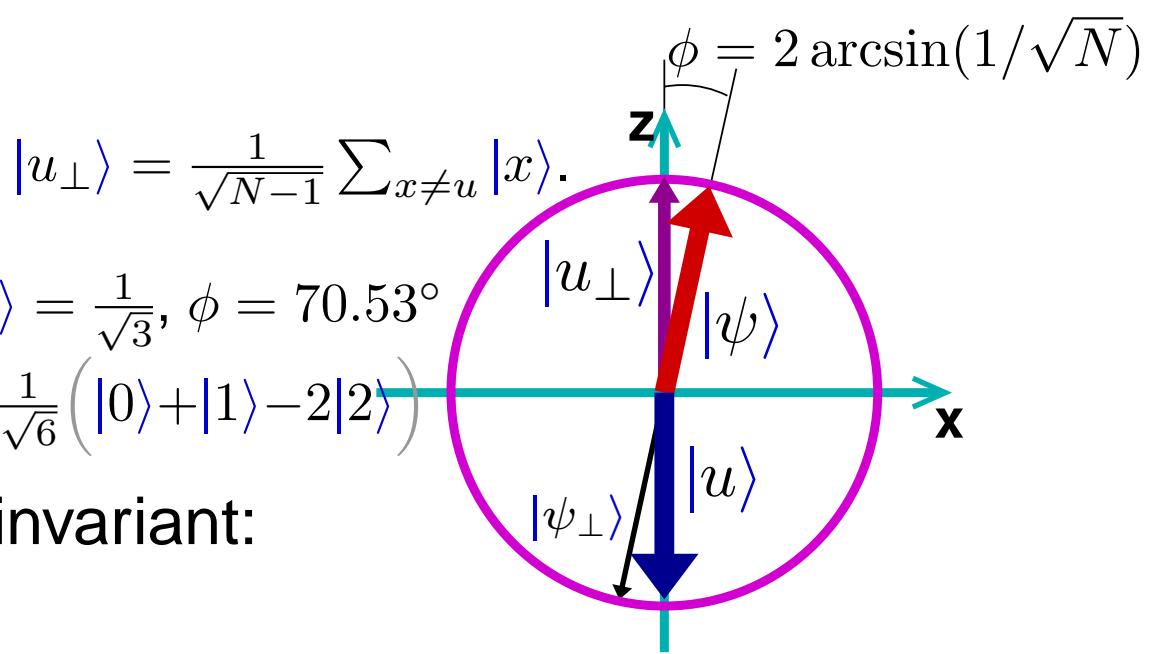
- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

- Operators that leave Q invariant:

- z BB. Acts as Z_{180° .



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

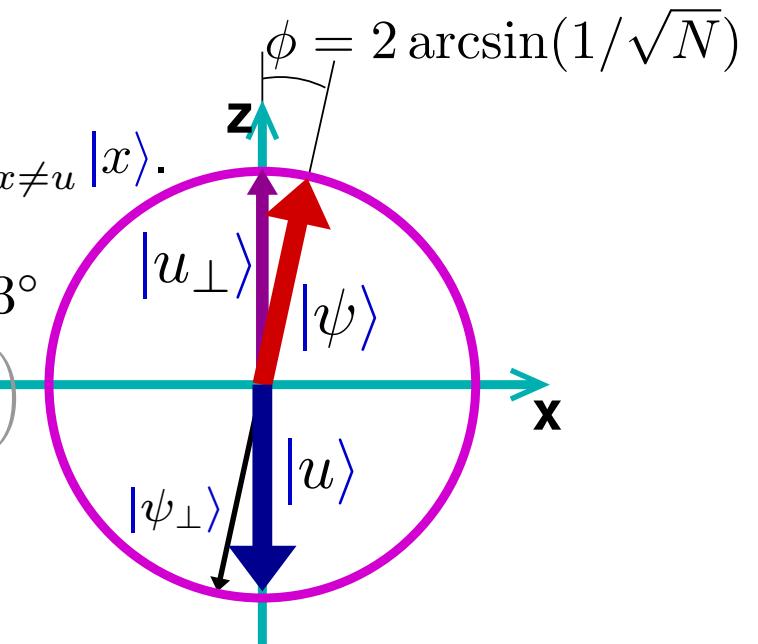
Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

$$|u_\perp\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq u} |x\rangle.$$



- Operators that leave Q invariant:

- z BB. Acts as Z_{180° .
- 180° rotation about $|\psi\rangle$:

$$\mathbf{H}\mathbf{Z}\mathbf{H}|\psi\rangle \rightarrow -|\psi\rangle$$

$$\mathbf{H}\mathbf{Z}\mathbf{H}|\psi_\perp\rangle \rightarrow |\psi_\perp\rangle \text{ if } \langle\psi|\psi_\perp\rangle = 0.$$



Grover's Algorithm: Rotations

- Rotate from $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ to $|u\rangle$.

Consider the 2-d subspace Q spanned by $|\psi\rangle$ and $|u\rangle$.

- Overlap: $\langle u|\psi\rangle = \frac{1}{\sqrt{N}}$.
- Bloch sphere picture:

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \quad \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \quad \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

- Operators that leave Q invariant:

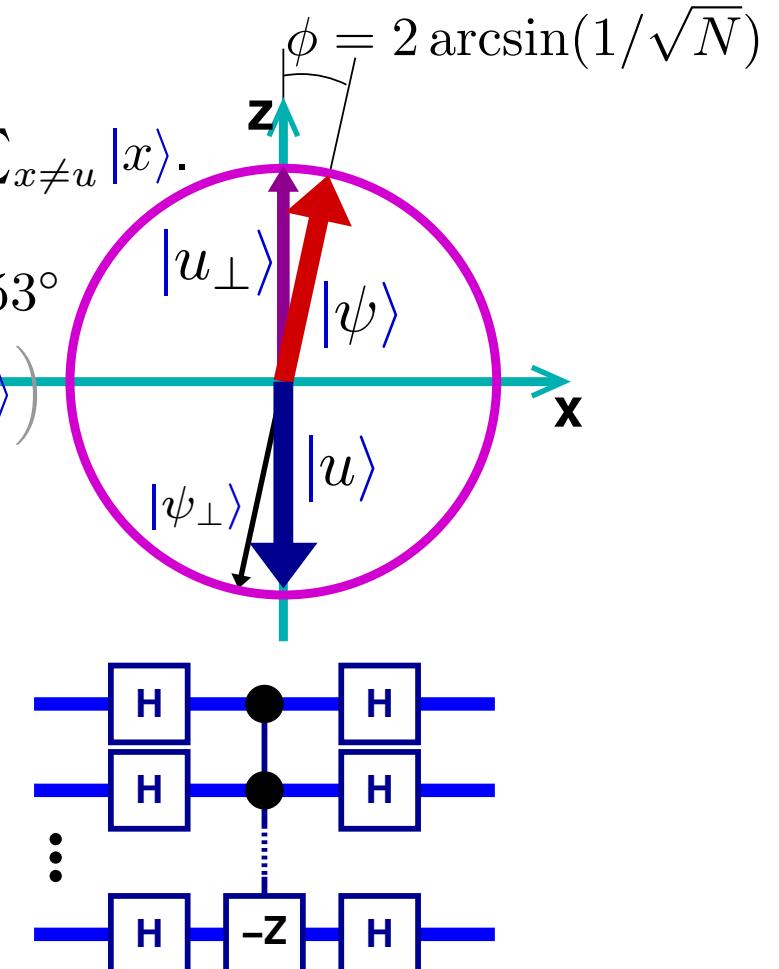
- z BB. Acts as Z_{180° .
- 180° rotation about $|\psi\rangle$:

$$HZH|\psi\rangle \rightarrow -|\psi\rangle$$

$$HZH|\psi_\perp\rangle \rightarrow |\psi_\perp\rangle \text{ if } \langle \psi|\psi_\perp\rangle = 0.$$

Qubit implementation of HZH :

$$|u_\perp\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq u} |x\rangle.$$



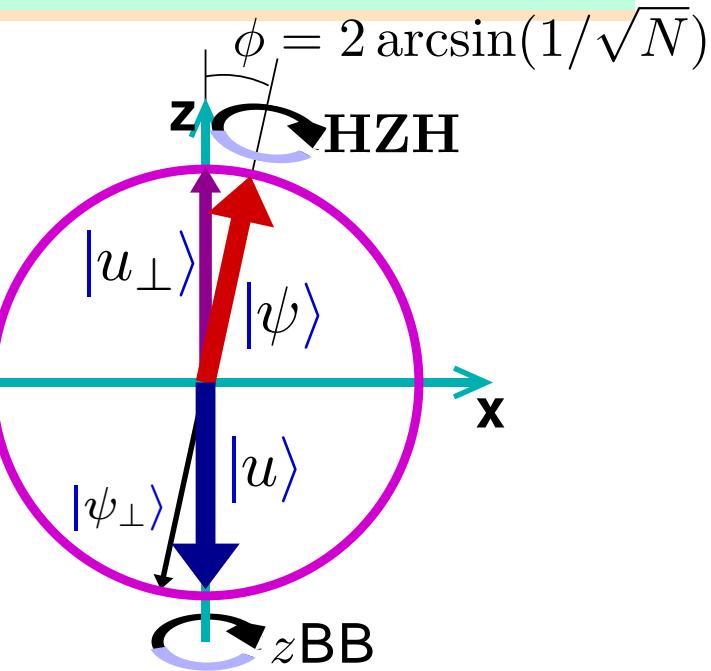
Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$



Grover's Algorithm

- Bloch sphere picture.

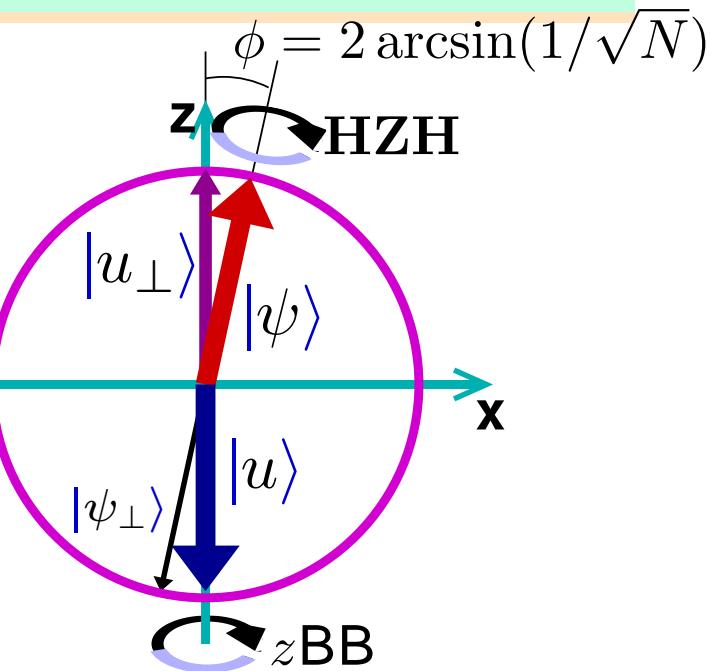
Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB}$$



Grover's Algorithm

- Bloch sphere picture.

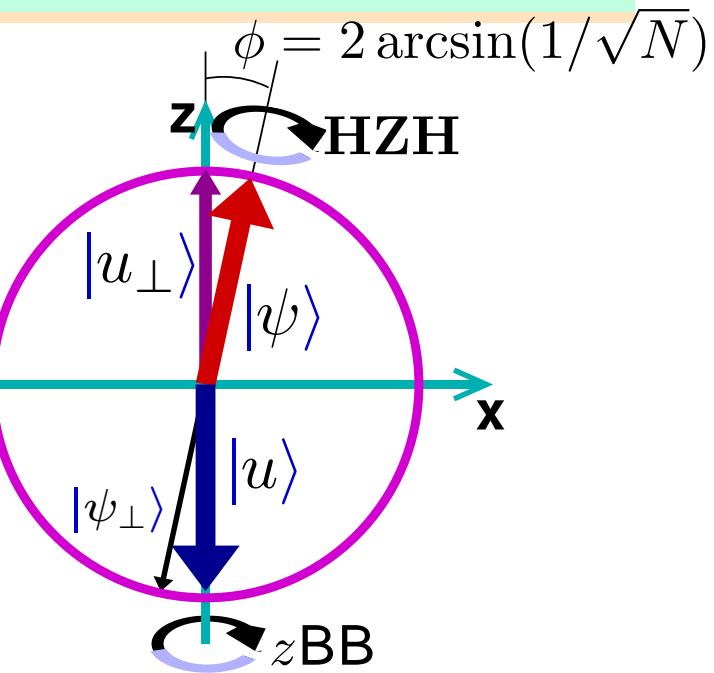
Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH}$$



Grover's Algorithm

- Bloch sphere picture.

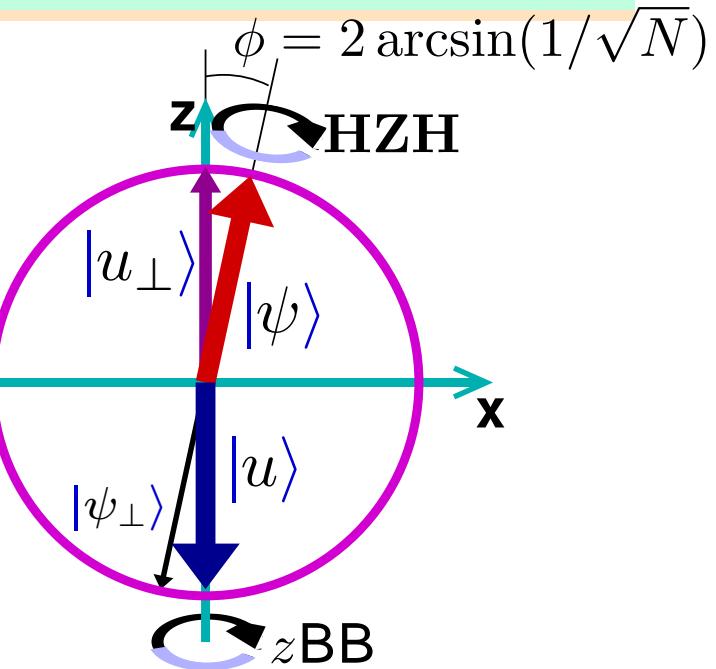
Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH} \hat{y}$$



Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

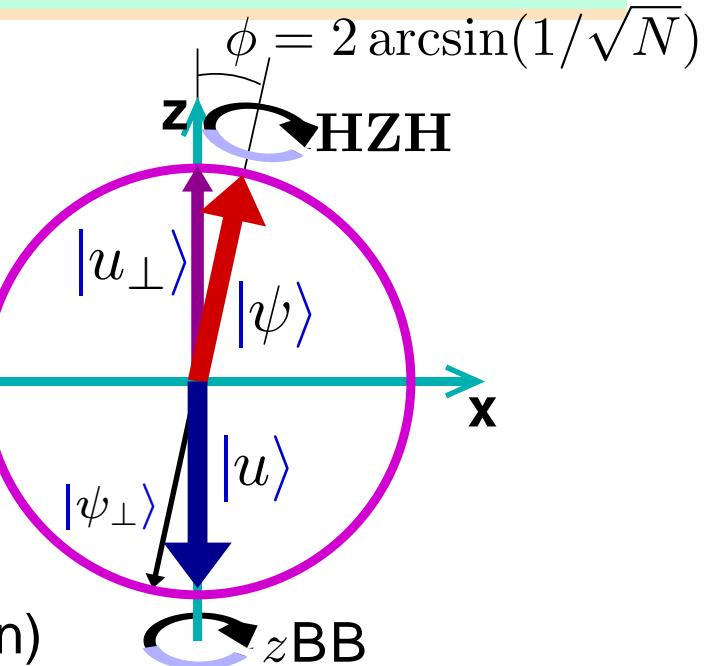
$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH} \hat{y}$$

(it is a y -rotation)



Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

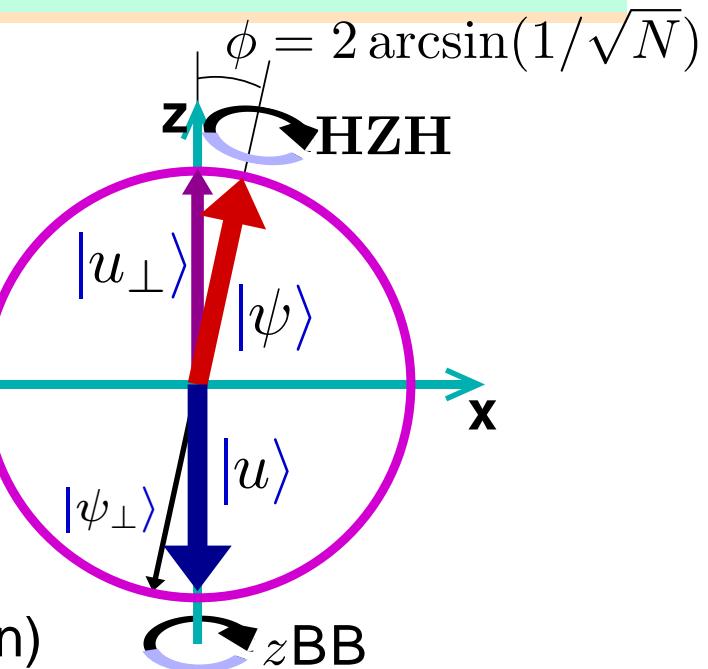
$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH} \hat{y}$$

(it is a y -rotation)

$$\hat{z} \xrightarrow{zBB}$$



Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

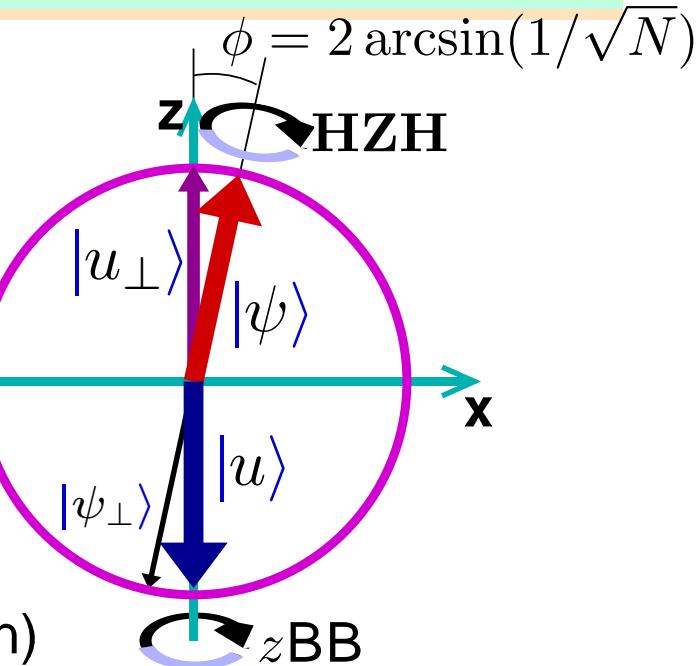
$$|u_{\perp}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_{\perp}\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of z BB.HZH in Bloch sphere:

$$\hat{y} \xrightarrow{z\mathbf{BB}} -\hat{y} \xrightarrow{\mathbf{HZH}} \hat{y}$$

(it is a y -rotation)

\hat{z} $\xrightarrow{z\text{BB}}$ \hat{z} $\xrightarrow{\text{HZH}}$



Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

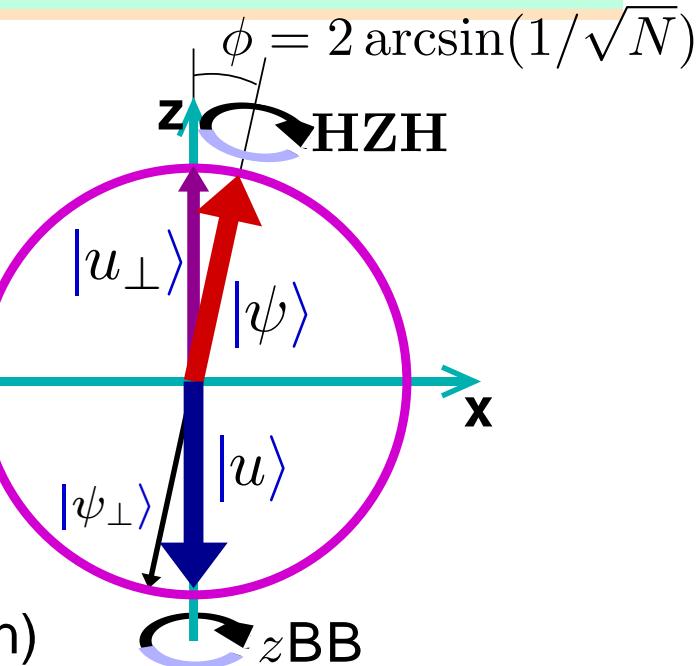
$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_{\perp}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_{\perp}\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of z BB.HZH in Bloch sphere:

$$\hat{y} \xrightarrow{z\text{BB}} -\hat{y} \xrightarrow{\text{HZH}} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \quad \xrightarrow{z\mathbf{B}\mathbf{B}} \quad \hat{z} \quad \xrightarrow{\mathbf{H}\mathbf{Z}\mathbf{H}} \quad \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$



Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

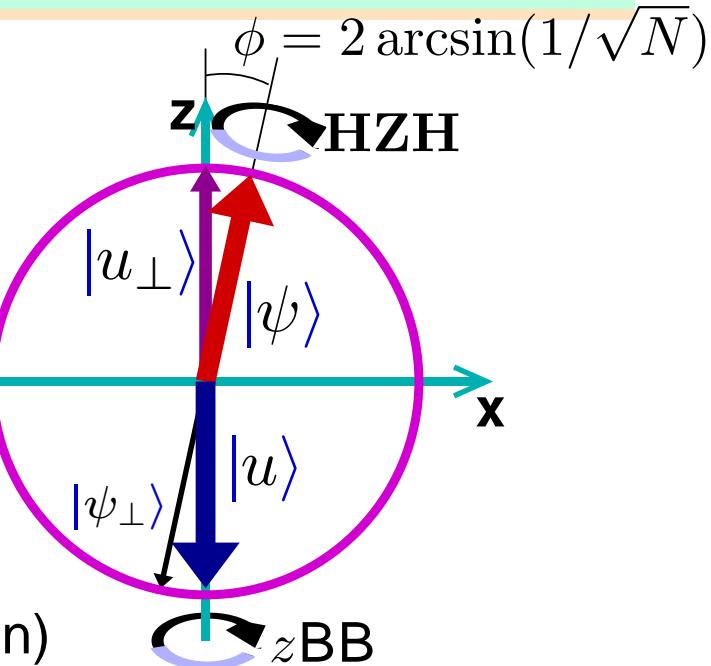
$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_{\perp}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_{\perp}\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of z BB.HZH in Bloch sphere:

$$\hat{y} \xrightarrow{z\text{BB}} -\hat{y} \xrightarrow{\text{HZH}} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \quad \xrightarrow{z\mathbf{B}\mathbf{B}} \quad \hat{z} \quad \xrightarrow{\mathbf{H}\mathbf{Z}\mathbf{H}} \quad \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$



(... by $4 \arcsin(1/\sqrt{N})$)

Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_{\perp}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_{\perp}\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

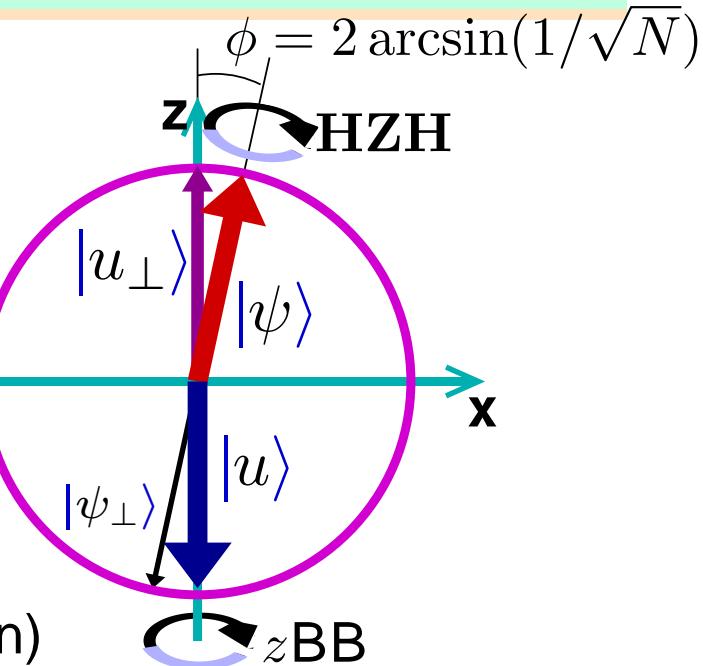
Effect of z BB.HZH in Bloch sphere:

$$\hat{y} \xrightarrow{z\text{BB}} -\hat{y} \xrightarrow{\text{HZH}} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \quad \xrightarrow{z\text{BB}} \quad \hat{z} \quad \xrightarrow{\text{HZH}} \quad \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$

- Grover's algorithm:

- # 1. Prepare $|\psi\rangle$.



(... by $4 \arcsin(1/\sqrt{N})$)

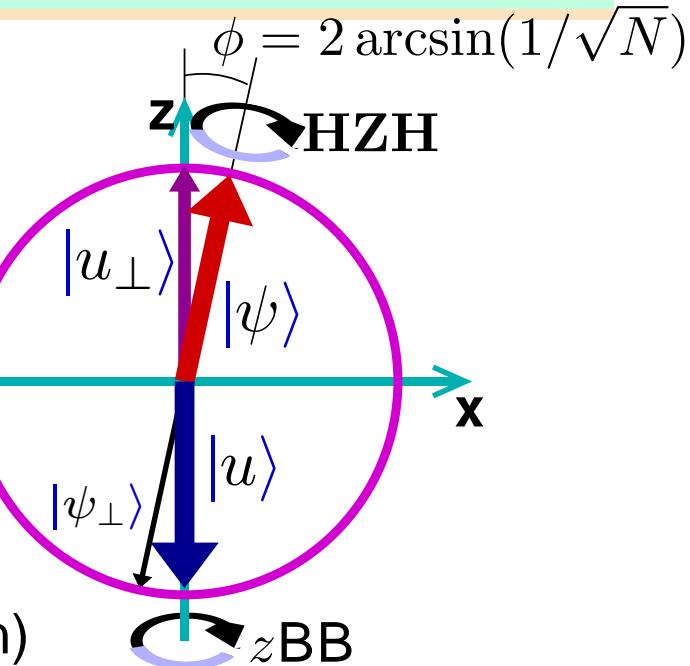
Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$



Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \xrightarrow{zBB} \hat{z} \xrightarrow{HZH} \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$

(... by $4 \arcsin(1/\sqrt{N})$)

- Grover's algorithm:

- Prepare $|\psi\rangle$.
- $(zBB.HZH)^{(\pi - 2 \arcsin(1/\sqrt{N})) / (4 \arcsin(1/\sqrt{N}))}$

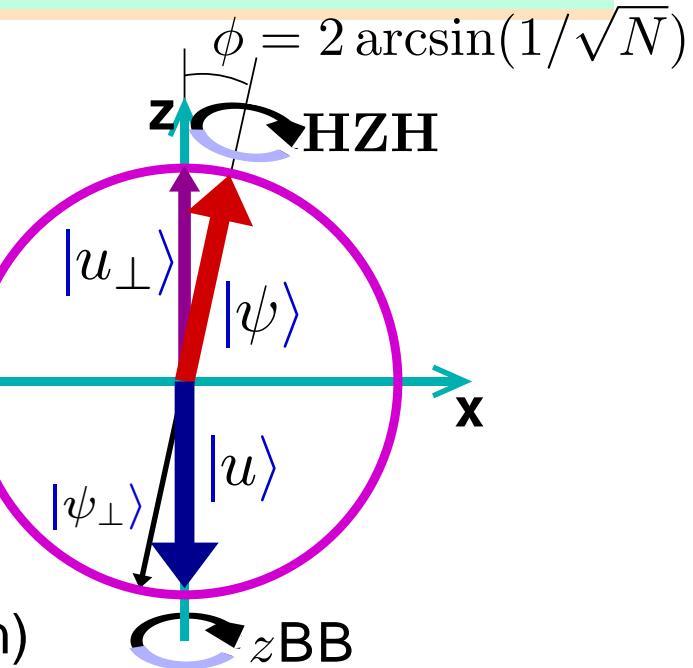
Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$



Effect of $zBB.HZH$ in Bloch sphere:

$$\hat{y} \xrightarrow{zBB} -\hat{y} \xrightarrow{HZH} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \xrightarrow{zBB} \hat{z} \xrightarrow{HZH} \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$

(... by $4 \arcsin(1/\sqrt{N})$)

- Grover's algorithm:

- Prepare $|\psi\rangle$.
- $(zBB.HZH)^{(\pi - 2 \arcsin(1/\sqrt{N})) / (4 \arcsin(1/\sqrt{N}))}$
- Measure logical basis.

Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

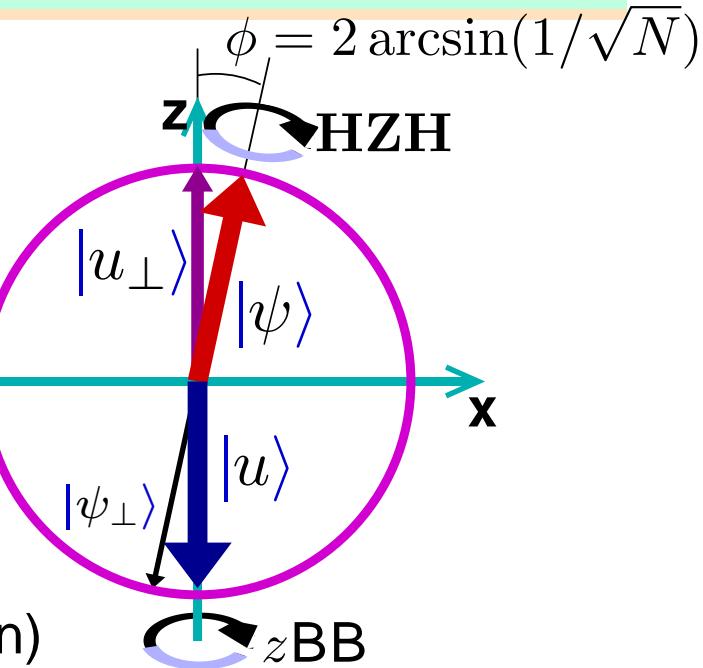
$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_{\perp}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_{\perp}\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$

Effect of z BB.HZH in Bloch sphere:

$$\hat{y} \xrightarrow{z\text{BB}} -\hat{y} \xrightarrow{\text{HZH}} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \quad \xrightarrow{z\text{BB}} \quad \hat{z} \quad \xrightarrow{\text{HZH}} \quad \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases}$$



(... by $4 \arcsin(1/\sqrt{N})$)

- Grover's algorithm:

1. Prepare $|\psi\rangle$.
 2. $(z\text{BB.HZH})^{(\pi - 2 \arcsin(1/\sqrt{N})) / (4 \arcsin(1/\sqrt{N}))}$
 3. Measure logical basis. ... rep

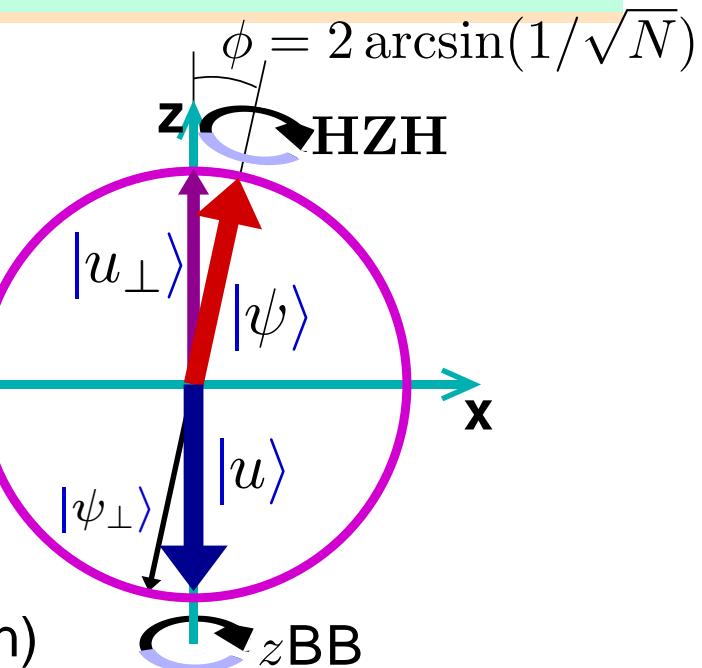
Grover's Algorithm

- Bloch sphere picture.

Example: $N = 3$, $|u\rangle = |2\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), \langle u|\psi\rangle = \frac{1}{\sqrt{3}}, \phi = 70.53^\circ$$

$$|u_\perp\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_\perp\rangle = \frac{1}{\sqrt{6}}(|0\rangle + |1\rangle - 2|2\rangle)$$



Effect of $z\text{BB}.\text{HZH}$ in Bloch sphere:

$$\hat{y} \xrightarrow{z\text{BB}} -\hat{y} \xrightarrow{\text{HZH}} \hat{y} \quad (\text{it is a } y\text{-rotation})$$

$$\hat{z} \xrightarrow{z\text{BB}} \hat{z} \xrightarrow{\text{HZH}} \begin{cases} \cos(4 \arcsin(1/\sqrt{N}))\hat{z} \\ + \sin(4 \arcsin(1/\sqrt{N}))\hat{x} \end{cases} \quad (\dots \text{by } 4 \arcsin(1/\sqrt{N}))$$

- Grover's algorithm:

- Prepare $|\psi\rangle$.

- $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(1/\sqrt{N})) / (4 \arcsin(1/\sqrt{N}))}$

- Measure logical basis.

... repeat, if necessary.

- Complexity: $\approx \pi\sqrt{N}/4$.



Quantum Database Search?

- An N -entry unstructured database is . . .



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (... sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (... sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.
- Quantum complexity with Grover’s algorithm. (. . . sequential)
Complexity of reversible $Q(\cdot)$: \tilde{q} . Q. access complexity: \tilde{a} .



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.
- Quantum complexity with Grover’s algorithm. (. . . sequential)
Complexity of reversible $Q(\cdot)$: \tilde{q} . Q. access complexity: \tilde{a} .
 - All items are accessed twice for each use of reversible Q .



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.
- Quantum complexity with Grover’s algorithm. (. . . sequential)
Complexity of reversible $Q(\cdot)$: \tilde{q} . Q. access complexity: \tilde{a} .
 - All items are accessed twice for each use of reversible Q .
 - Q may have to be reversibly computed twice in each iteration.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.
- Quantum complexity with Grover’s algorithm. (. . . sequential)
Complexity of reversible $Q(\cdot)$: \tilde{q} . Q. access complexity: \tilde{a} .
 - All items are accessed twice for each use of reversible Q .
 - Q may have to be reversibly computed twice in each iteration.
 - Total complexity: $\Omega(\sqrt{N}(2N\tilde{a} + \tilde{q}))$.



Quantum Database Search?

- An N -entry unstructured database is . . .
 N items $D(i)$ stored at classical memory locations $1, \dots, N$.
- A generic query: “Return an index i such that $Q(D(i)) = 1$.
 $Q(\cdot)$ is a subroutine provided with the query.
- Classical complexity for unique answers. (. . . sequential)
Complexity of $Q(\cdot)$: q . Item access complexity: a .
 - On average, half the items must be accessed.
 - The query function is executed for each item accessed.
 - Total complexity: $O(N(a + q)/2)$.
- Quantum complexity with Grover’s algorithm. (. . . sequential)
Complexity of reversible $Q(\cdot)$: \tilde{q} . Q. access complexity: \tilde{a} .
 - All items are accessed twice for each use of reversible Q .
 - Q may have to be reversibly computed twice in each iteration.
 - Total complexity: $\Omega(\sqrt{N}(2N\tilde{a} + \tilde{q}))$.Grover can beat classical only if $q \gg N^{1/2}\tilde{a}$.



Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .

Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.



Unstructured Quantum Search

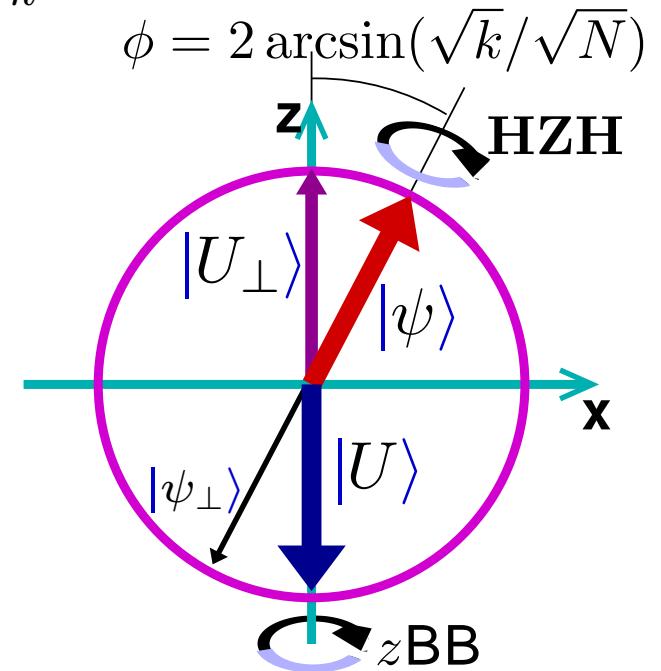
- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.



Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.

$$|U_\perp\rangle = \frac{1}{\sqrt{N-k}} \sum_{x \notin U} |x\rangle$$



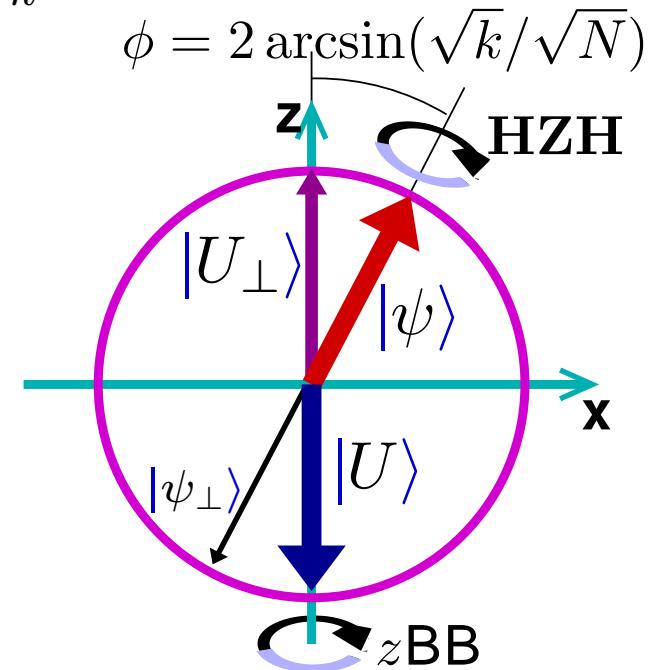
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .

- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{|x \in U|}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.

$$|U_\perp\rangle = \frac{1}{\sqrt{N-k}} \sum_{x \notin U} |x\rangle$$

$$z\text{BB}|U\rangle = -|U\rangle, z\text{BB}|U_\perp\rangle = |U_\perp\rangle$$



Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.

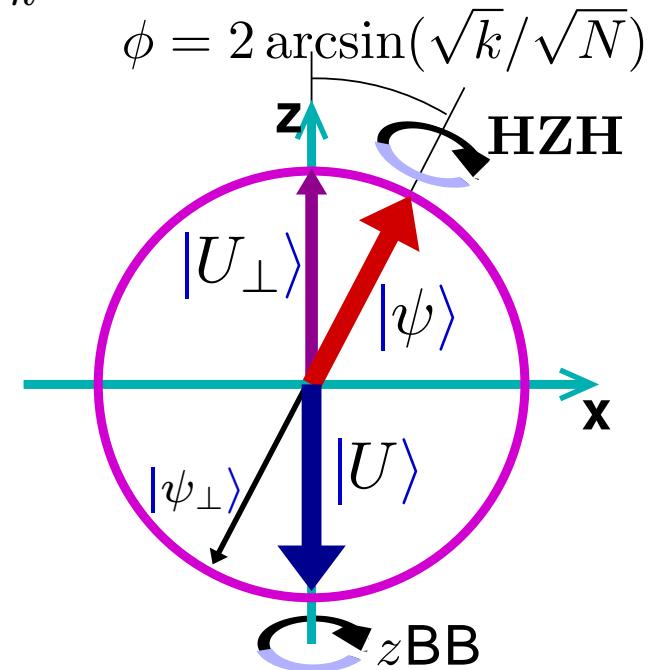
1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{|x \in U|}|x\rangle$ by phase kickback.

– $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.

$$|U_\perp\rangle = \frac{1}{\sqrt{N-k}} \sum_{x \notin U} |x\rangle$$

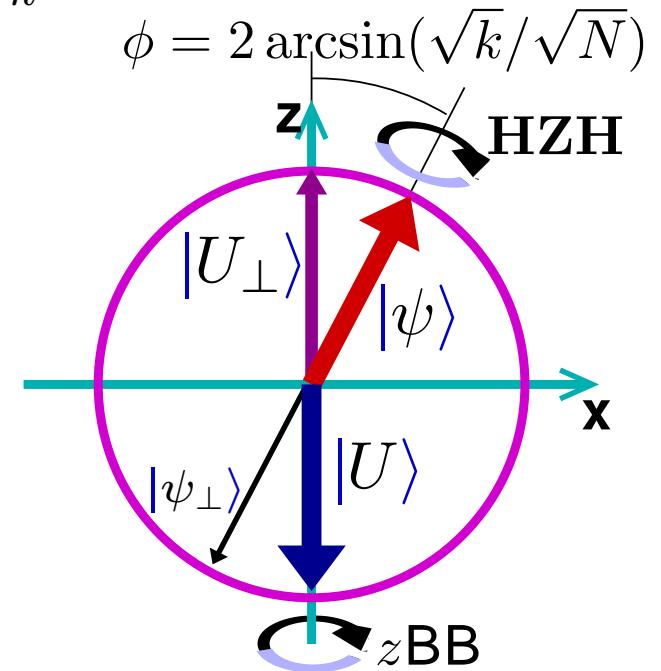
$$z\text{BB}|U\rangle = -|U\rangle, z\text{BB}|U_\perp\rangle = |U_\perp\rangle$$

$$\text{HZH}|\psi\rangle = -|\psi\rangle, \text{HZH}|\psi_\perp\rangle = |\psi_\perp\rangle$$



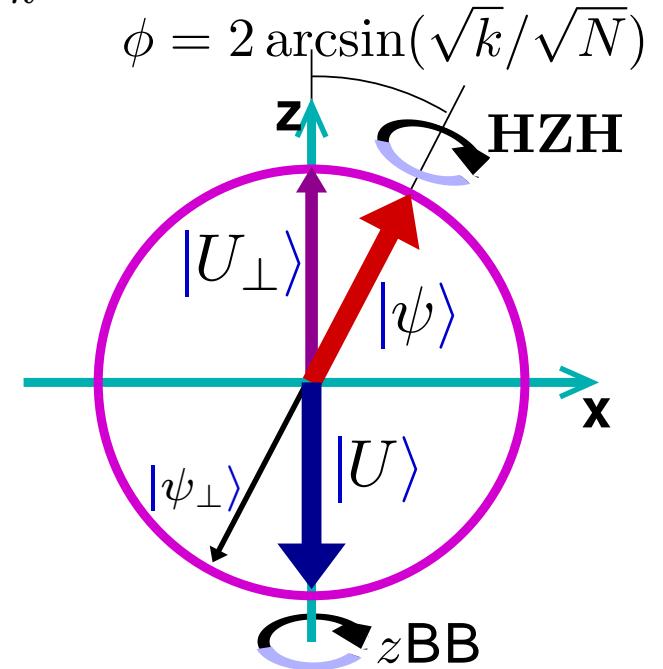
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{|x \in U|}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.



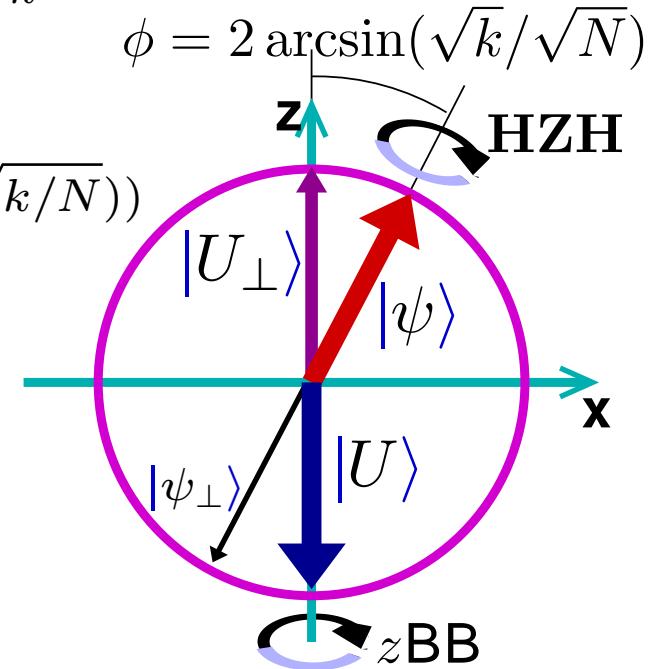
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 1. Construct $z\text{BB} : |x\rangle \mapsto (-1)^{|x \in U|}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 2. Prepare $|\psi\rangle$



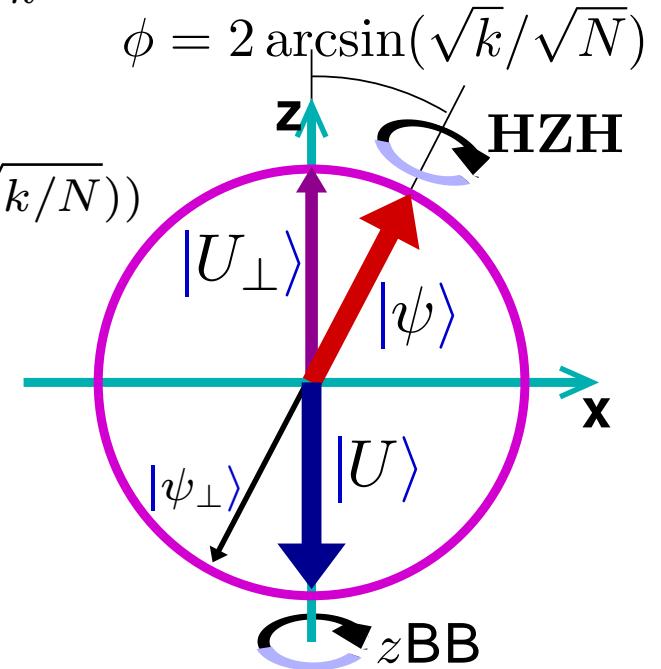
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 - Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 - Prepare $|\psi\rangle$
 - $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(\sqrt{k}/\sqrt{N}))/(4 \arcsin(\sqrt{k}/\sqrt{N}))}$



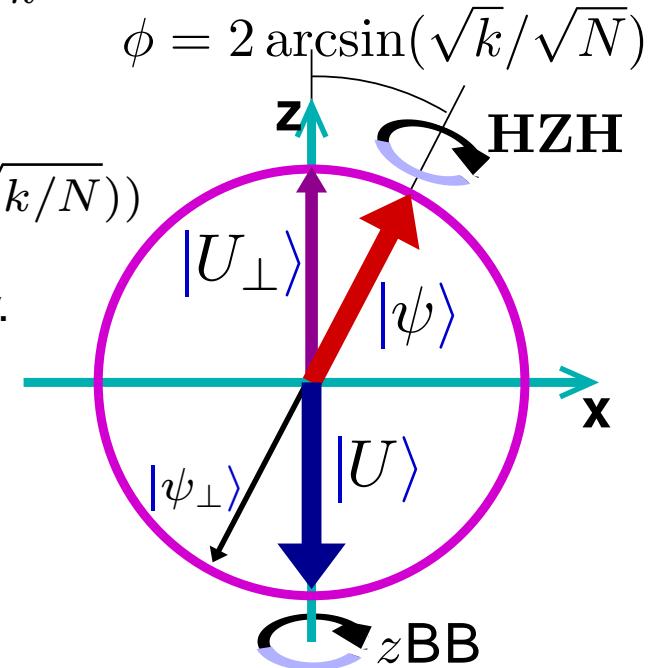
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 - Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 - Prepare $|\psi\rangle$
 - $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(\sqrt{k}/\sqrt{N}))/(4 \arcsin(\sqrt{k}/\sqrt{N}))}$
 - Measure logical basis.



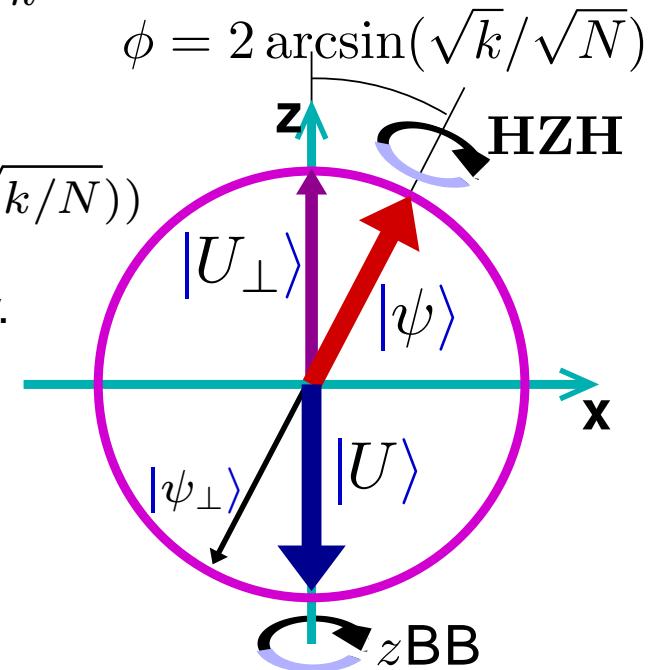
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 - Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 - Prepare $|\psi\rangle$
 - $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(\sqrt{k}/\sqrt{N}))/(4 \arcsin(\sqrt{k}/\sqrt{N}))}$
 - Measure logical basis. ... repeat, if necessary.



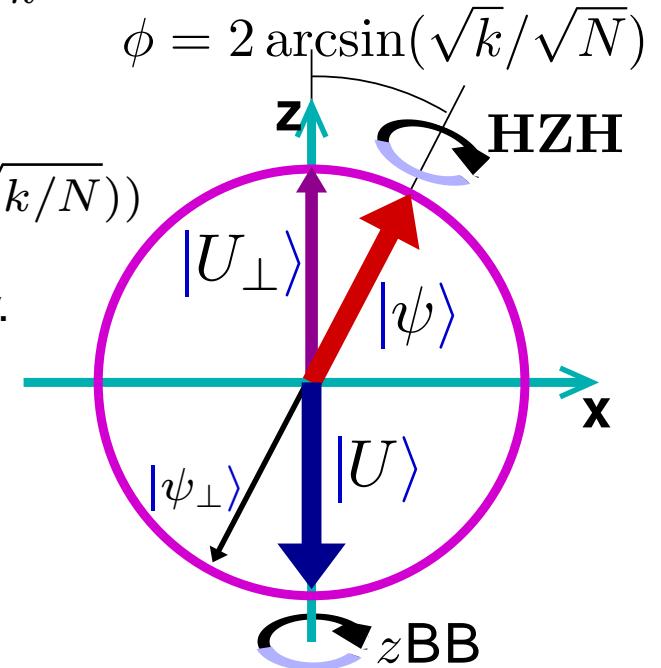
Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 - Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 - Prepare $|\psi\rangle$
 - $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(\sqrt{k}/\sqrt{N}))/(4 \arcsin(\sqrt{k}/\sqrt{N}))}$
 - Measure logical basis. ... repeat, if necessary.
- Complexity: $\approx \pi \sqrt{N/k}/4$.



Unstructured Quantum Search

- Given: BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$, $|U| = k$.
Problem: Find an element of U .
- Algorithm.
 - Construct $z\text{BB} : |x\rangle \mapsto (-1)^{[x \in U]}|x\rangle$ by phase kickback.
 - $z\text{BB}$ and HZH preserve $\text{span}(|U\rangle = \frac{1}{\sqrt{k}} \sum_{x \in U} |x\rangle, |\psi\rangle)$.
 - Prepare $|\psi\rangle$
 - $(z\text{BB}.\text{HZH})^{(\pi - 2 \arcsin(\sqrt{k}/\sqrt{N}))/(4 \arcsin(\sqrt{k}/\sqrt{N}))}$
 - Measure logical basis. ... repeat, if necessary.
- Complexity: $\approx \pi \sqrt{N/k}/4$.
- If k is unknown: Binary search on k .
Try $k=N/2, k=N/4, k=N/8, \dots$
Check solutions.



Quantum Counting

- Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
- Problem: Determine $|U|/N$ to within ϵ .
... let $k = |U|$.



Quantum Counting

"implementable as a quantum controlled operation"

- Given: (c-)BB such that $\text{BB} |x\rangle_S |b\rangle_T = |x\rangle_S |b+ [x \in U]\rangle_T$.
- Problem: Determine $|U|/N$ to within ϵ .
... let $k = |U|$.



Quantum Counting

"implementable as a quantum controlled operation"

- Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine $|U|/N$ to within ϵ . . . let $k = |U|$.
- A Grover iterate $z\text{BB.HZH}$ is a Bloch-sphere rotation by $4 \arcsin(\sqrt{k/N})$ in the 2-d space containing $|\psi\rangle$ and $|U\rangle$.



Quantum Counting

"implementable as a quantum controlled operation"

- Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine $|U|/N$ to within ϵ . . . let $k = |U|$.
- A Grover iterate $z\text{BB.HZH}$ is a Bloch-sphere rotation by $4 \arcsin(\sqrt{k/N})$ in the 2-d space containing $|\psi\rangle$ and $|U\rangle$.
 - Idea: Measure an eigenvalue of $z\text{BB.HZH}$.
The eigenvalues are
$$-e^{\pm 2 \arcsin(\sqrt{k/N})i} = -(\sqrt{(N-k)/N}) \pm i\sqrt{k/N})^2.$$

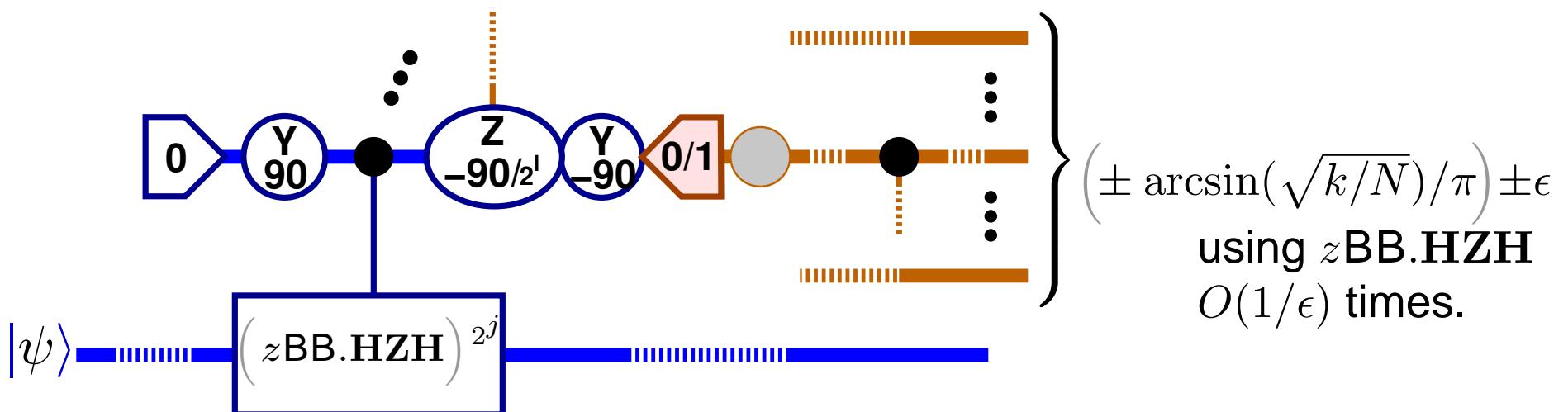


Quantum Counting

"implementable as a quantum controlled operation"

- Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine $|U|/N$ to within ϵ . . . let $k = |U|$.
- A Grover iterate $z\text{BB.HZH}$ is a Bloch-sphere rotation by $4 \arcsin(\sqrt{k/N})$ in the 2-d space containing $|\psi\rangle$ and $|U\rangle$.
 - Idea: Measure an eigenvalue of $z\text{BB.HZH}$.
The eigenvalues are

$$-e^{\pm 2 \arcsin(\sqrt{k/N})i} = -(\sqrt{(N-k)/N}) \pm i\sqrt{k/N})^2.$$



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.

- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.

- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .
1. Randomly choose l distinct elements. r is the fraction in U .



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.

- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .

1. Randomly choose l distinct elements. r is the fraction in U .

$$2. \langle r \rangle = u. \text{ std}(r) = \sqrt{u(1-u)(1-\frac{l-1}{n-1})/l}.$$



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

$$1. \frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1.$$

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.

- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .

1. Randomly choose l distinct elements. r is the fraction in U .

2. $\langle r \rangle = u$. $\text{std}(r) = \sqrt{u(1-u)(1-\frac{l-1}{n-1})/l}$.

3. So set $l > u(1-u)(1-\frac{l-1}{n-1})/\epsilon^2 \approx u(1-u)/\epsilon^2$ for $l \ll n$.



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .

1. $\frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1$.

2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.

3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.

- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .

1. Randomly choose l distinct elements. r is the fraction in U .

2. $\langle r \rangle = u$. $\text{std}(r) = \sqrt{u(1-u)(1-\frac{l-1}{n-1})/l}$.

3. So set $l > u(1-u)(1-\frac{l-1}{n-1})/\epsilon^2 \approx u(1-u)/\epsilon^2$ for $l \ll n$.

4. Effort required: $l = O(u(1-u)/\epsilon^2)$ for $l \ll n, u > 0$.



Quantum versus Classical Counting

Let $u = |U|/N$.

- Quantum: Given: (c-)BB such that $\text{BB}|x\rangle_S|b\rangle_T = |x\rangle_S|b+[x \in U]\rangle_T$.
Problem: Determine u to within ϵ .
 1. $\frac{d}{dt} \arcsin(\sqrt{t}) = \frac{1}{2\sqrt{t(1-t)}} \geq 1$.
 2. Determine $\arcsin(\sqrt{u})$ within $\delta = \epsilon/(2\sqrt{(u+\epsilon)(1-u+\epsilon)})$.
 3. Effort required: $O(\sqrt{(u+\epsilon)(1-u+\epsilon)}/\epsilon)$ uses of zBB.HZH.
- Classical: Given: BB such that $\text{BB}(x) = [x \in U]$.
Problem: Determine u to within ϵ .
 1. Randomly choose l distinct elements. r is the fraction in U .
 2. $\langle r \rangle = u$. $\text{std}(r) = \sqrt{u(1-u)(1-\frac{l-1}{n-1})/l}$.
 3. So set $l > u(1-u)(1-\frac{l-1}{n-1})/\epsilon^2 \approx u(1-u)/\epsilon^2$ for $l \ll n$.
 4. Effort required: $l = O(u(1-u)/\epsilon^2)$ for $l \ll n, u > 0$.
- With these methods: Quantum counting is quadratically more efficient than classical probabilistic counting.



Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|0\rangle$ is not in the $+1$ eigenspace of zBB .
Problem: Prepare a state in the -1 eigenspace of zBB .



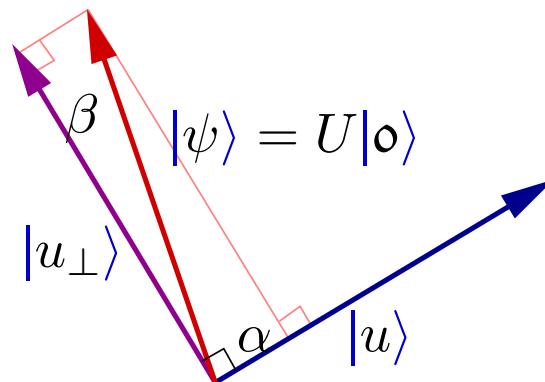
Algorithms for Amplitude

- Given: $z\mathbf{B}\mathbf{B}$ with eigenvalues in $\{-1, +1\}$ and U such that $U|\text{o}\rangle$ is not in the $+1$ eigenspace of $z\mathbf{B}\mathbf{B}$.
Problem: Prepare a state in the -1 eigenspace of $z\mathbf{B}\mathbf{B}$.
- Write $U|\text{o}\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $z\mathbf{B}\mathbf{B}|u\rangle = -|u\rangle$, $z\mathbf{B}\mathbf{B}|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.



Algorithms for Amplitude

- Given: $z\text{BB}$ with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of $z\text{BB}$.
Problem: Prepare a state in the -1 eigenspace of $z\text{BB}$.
 - Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $z\text{BB}|u\rangle = -|u\rangle$, $z\text{BB}|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.
- Hilbert space picture:



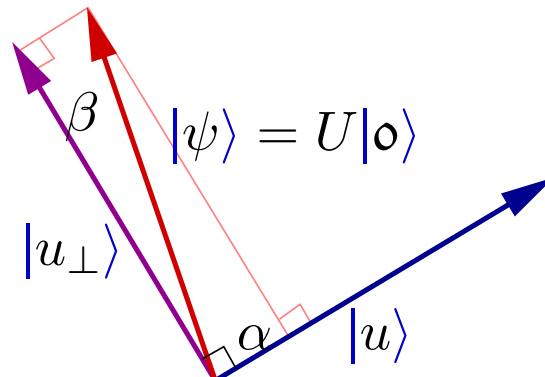
Algorithms for Amplitude

- Given: $z\text{BB}$ with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of $z\text{BB}$.

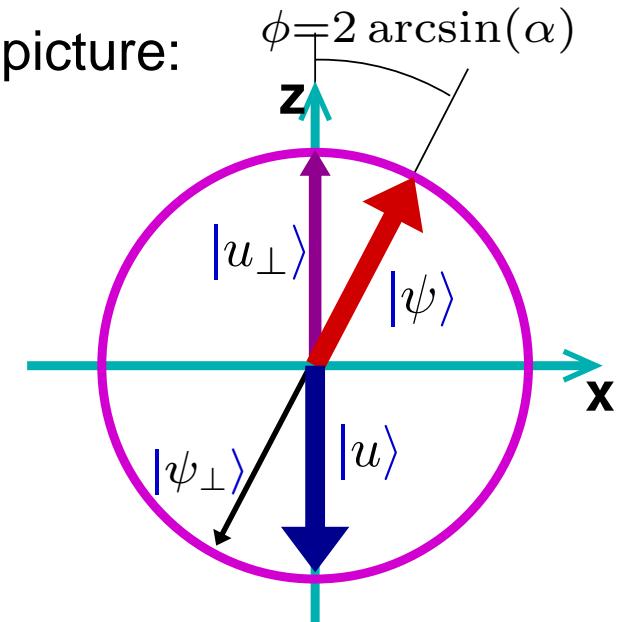
Problem: Prepare a state in the -1 eigenspace of $z\text{BB}$.

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $z\text{BB}|u\rangle = -|u\rangle$, $z\text{BB}|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

Hilbert space picture:



Bloch sphere picture:



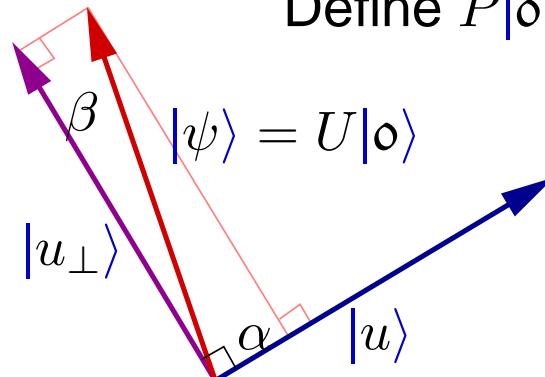
Algorithms for Amplitude

- Given: $z\text{BB}$ with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of $z\text{BB}$.

Problem: Prepare a state in the -1 eigenspace of $z\text{BB}$.

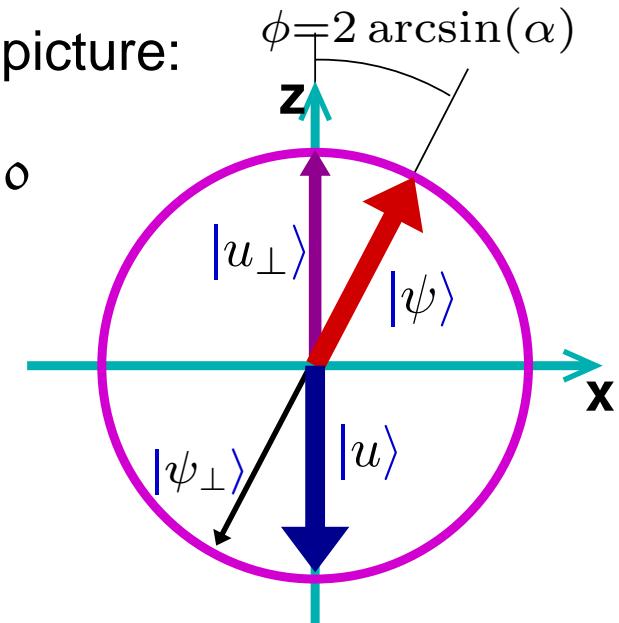
- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $z\text{BB}|u\rangle = -|u\rangle$, $z\text{BB}|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

Hilbert space picture:



Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

Bloch sphere picture:



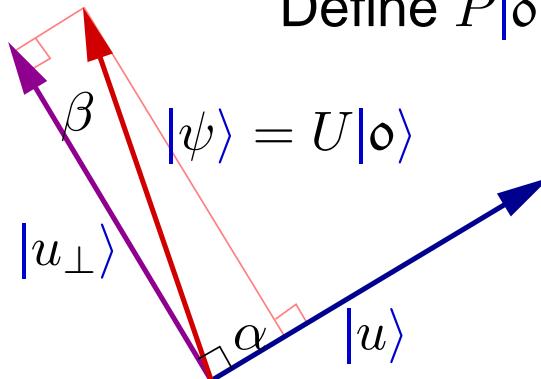
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

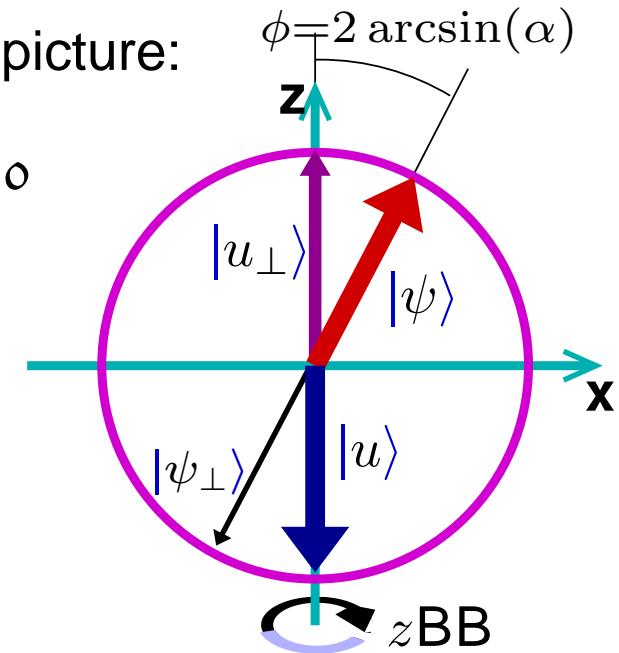
- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

Hilbert space picture:



Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

Bloch sphere picture:



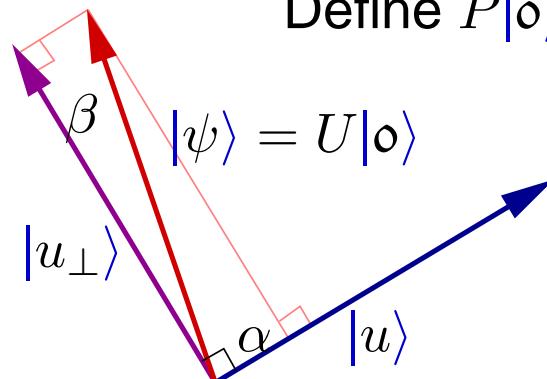
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

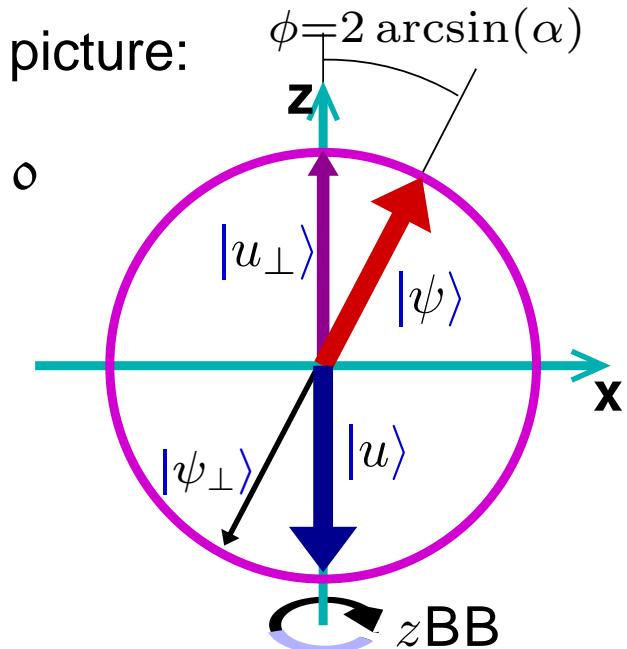
Hilbert space picture:



Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_{\perp}\rangle &= |\psi_{\perp}\rangle \end{aligned}$$

Bloch sphere picture:



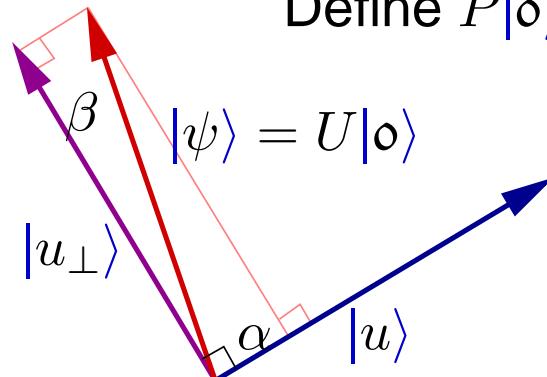
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

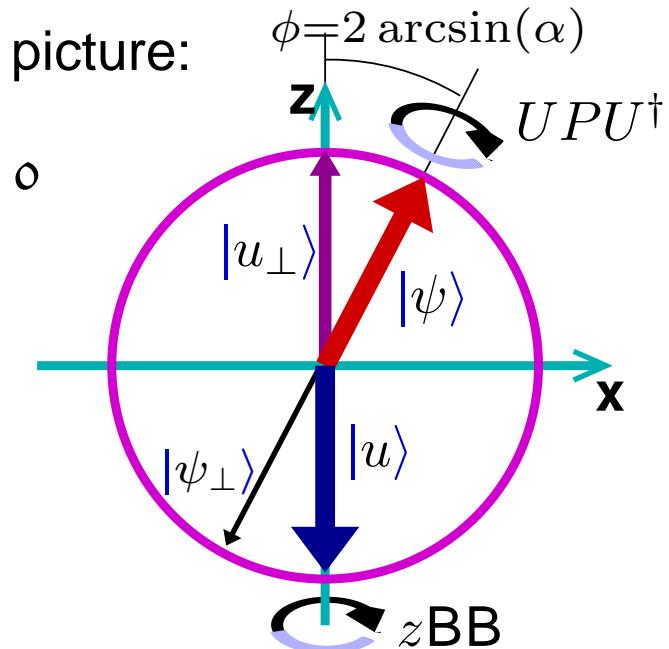
Hilbert space picture:



Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_{\perp}\rangle &= |\psi_{\perp}\rangle \end{aligned}$$

Bloch sphere picture:



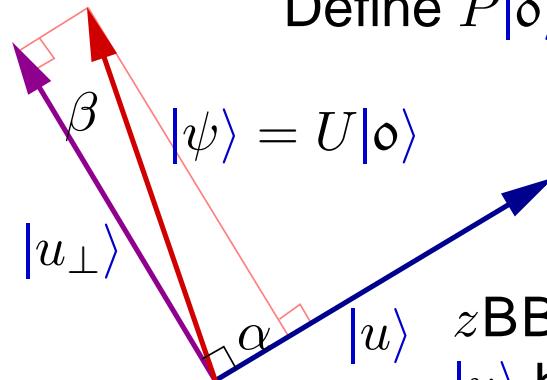
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.

Hilbert space picture:

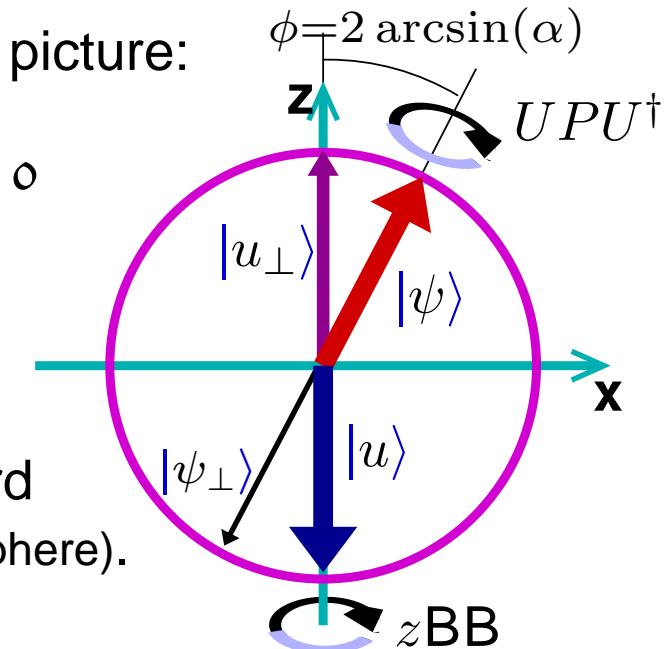


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_\perp\rangle &= |\psi_\perp\rangle \end{aligned}$$

$zBB.(UPU^\dagger)$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4\arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



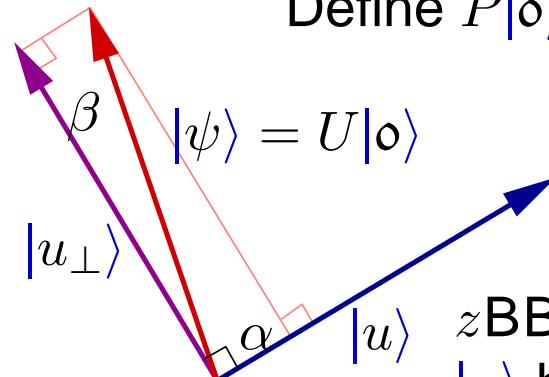
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.

Hilbert space picture:

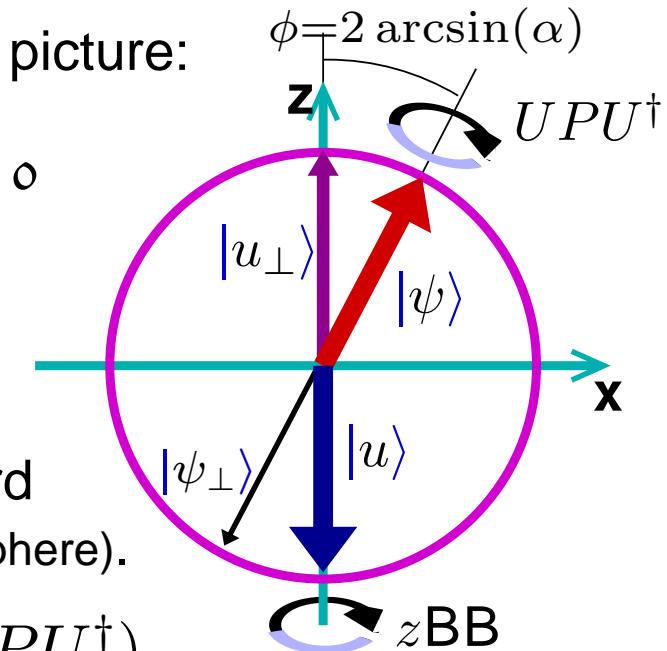


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_\perp\rangle &= |\psi_\perp\rangle \end{aligned}$$

$zBB.(UPU^\dagger)$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4\arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



- “Amplify” overlap of $|\psi\rangle$ with $|u\rangle$ by $zBB.(UPU^\dagger)$.



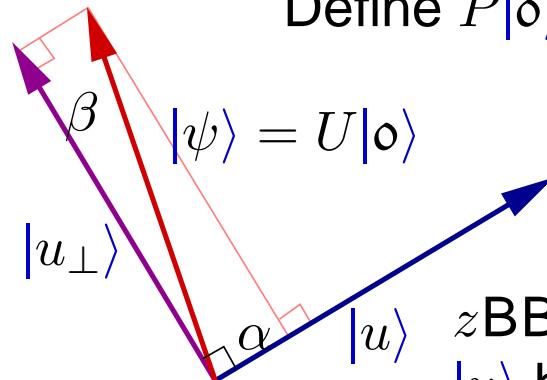
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.

Hilbert space picture:

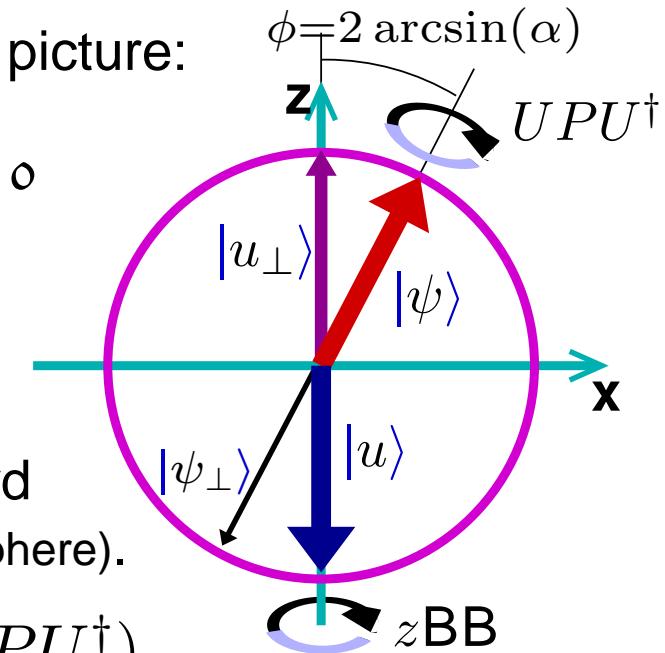


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_\perp\rangle &= |\psi_\perp\rangle \end{aligned}$$

$zBB.(UPU^\dagger)$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4 \arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



- “Amplify” overlap of $|\psi\rangle$ with $|u\rangle$ by $zBB.(UPU^\dagger)$.
 $(zBB.(UPU^\dagger))^{\pi/(4 \arcsin(\alpha)) - 1/2}|\psi\rangle$ to come closest to $|u\rangle$.



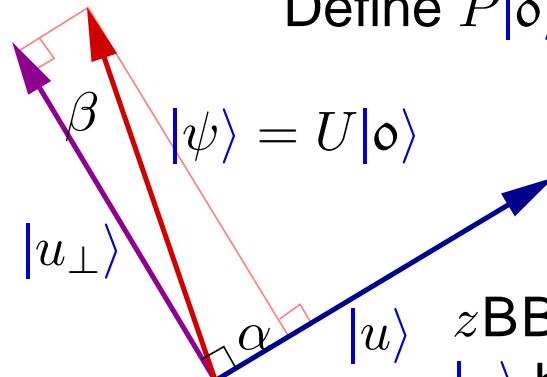
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_{\perp}\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_{\perp}\rangle = |u_{\perp}\rangle$, α, β non-negative real.

Hilbert space picture:

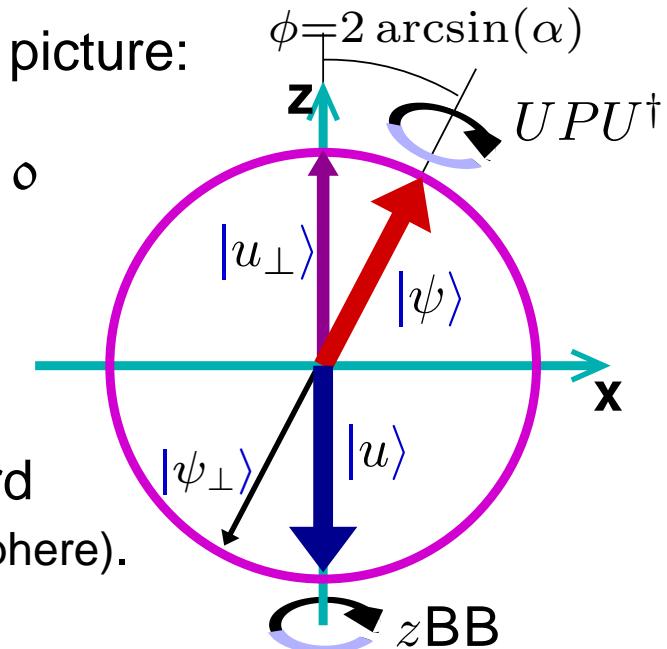


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^{\dagger}|\psi\rangle &= -|\psi\rangle \\ UPU^{\dagger}|\psi_{\perp}\rangle &= |\psi_{\perp}\rangle \end{aligned}$$

$zBB.(UPU^{\dagger})$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4 \arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



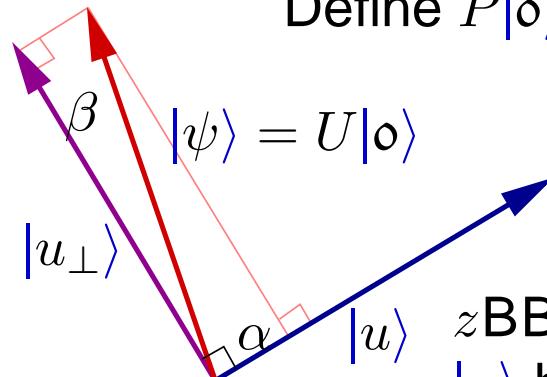
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.

Hilbert space picture:

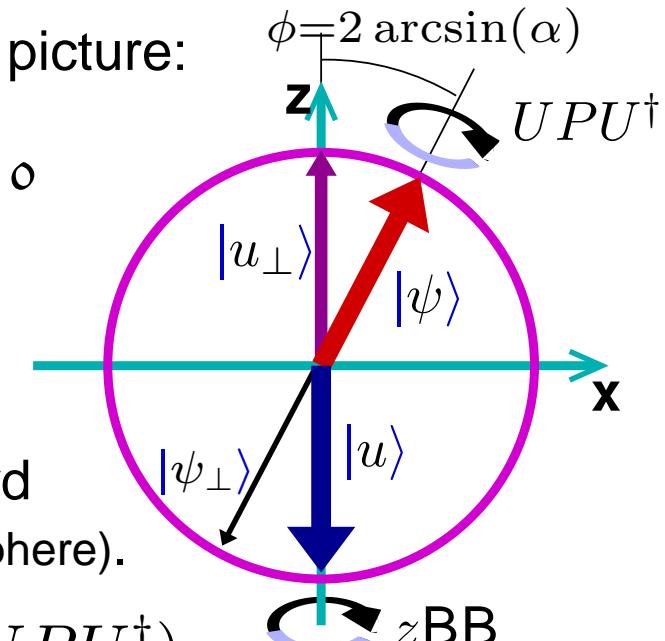


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_\perp\rangle &= |\psi_\perp\rangle \end{aligned}$$

$zBB.(UPU^\dagger)$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4\arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



- “Estimate” overlap of $|\psi\rangle$ with $|u\rangle$ by $zBB.(UPU^\dagger)$.



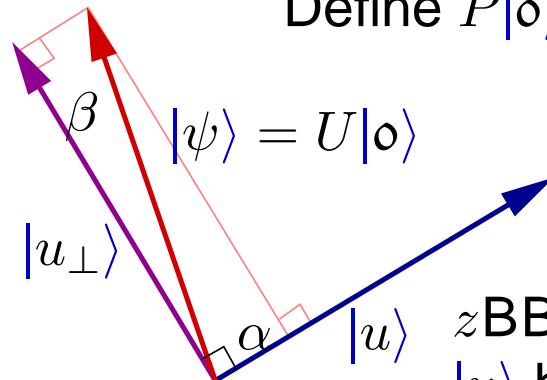
Algorithms for Amplitude

- Given: zBB with eigenvalues in $\{-1, +1\}$ and U such that $U|\phi\rangle$ is not in the $+1$ eigenspace of zBB .

Problem: Prepare a state in the -1 eigenspace of zBB .

- Write $U|\phi\rangle = \alpha|u\rangle + \beta|u_\perp\rangle$,
with $zBB|u\rangle = -|u\rangle$, $zBB|u_\perp\rangle = |u_\perp\rangle$, α, β non-negative real.

Hilbert space picture:

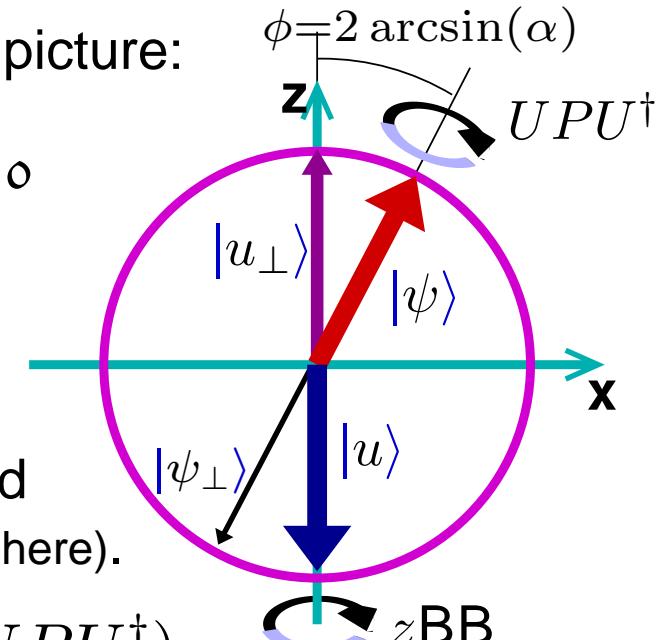


Define $P|\phi\rangle = -|\phi\rangle$, $P|b\rangle = |b\rangle$ for $b \neq \phi$

$$\begin{aligned} UPU^\dagger|\psi\rangle &= -|\psi\rangle \\ UPU^\dagger|\psi_\perp\rangle &= |\psi_\perp\rangle \end{aligned}$$

$zBB.(UPU^\dagger)$ rotates $|\psi\rangle$ toward $|u\rangle$ by $4\arcsin(\alpha)$ (in the Bloch sphere).

Bloch sphere picture:



- “Estimate” overlap of $|\psi\rangle$ with $|u\rangle$ by $zBB.(UPU^\dagger)$.
Measure an eigenv. of $zBB.(UPU^\dagger)$ on $|\psi\rangle$, get $\arcsin(\alpha) \pm \epsilon$.



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
- Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm.
 1. Choose k random, distinct inputs x_1, \dots, x_k .



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm.
 1. Choose k random, distinct inputs x_1, \dots, x_k .
 2. Compute $E_k = \frac{1}{k} \sum_j f(x_j)$.



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
 - Classical probabilistic algorithm.
 - Choose k random, distinct inputs x_1, \dots, x_k .
 - Compute $E_k = \frac{1}{k} \sum_j f(x_j)$.
- Properties: $\langle E_k \rangle = \langle f \rangle$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm.
 - Choose k random, distinct inputs x_1, \dots, x_k .
 - Compute $E_k = \frac{1}{k} \sum_j f(x_j)$.

Properties: $\langle E_k \rangle = \langle f \rangle$
 $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm.
 - Choose k random, distinct inputs x_1, \dots, x_k .
 - Compute $E_k = \frac{1}{k} \sum_j f(x_j)$.
- Properties: $\langle E_k \rangle = \langle f \rangle$
 $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Applications.
 - Numerical integration in many dimensions.
 - Monte Carlo path integration.



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .

- Classical probabilistic algorithm.
 - Choose k random, distinct inputs x_1, \dots, x_k .
 - Compute $E_k = \frac{1}{k} \sum_j f(x_j)$.

Properties: $\langle E_k \rangle = \langle f \rangle$
 $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$

- Applications.
 - Numerical integration in many dimensions.
 - Monte Carlo path integration.
- Goal. Double the number of digits of precision for similar effort.



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\mathbf{B}\mathbf{B}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{0}\rangle \right) = U_f |0\rangle |\mathbf{0}\rangle$$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\mathbf{BB}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{0}\rangle \right) = U_f |0\rangle |\mathbf{0}\rangle$$

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{f(x)/M} |\mathbf{1}\rangle, \quad |u_{\perp}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{1-f(x)/M} |\mathbf{0}\rangle$$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\mathbf{B}\mathbf{B}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{o}\rangle \right) = U_f |0\rangle |\mathbf{o}\rangle$$

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{f(x)/M} |\mathbf{1}\rangle, \quad |u_{\perp}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{1-f(x)/M} |\mathbf{o}\rangle$$

$$\alpha = \langle u | U_f | 0 \rangle |\mathbf{o}\rangle = \frac{1}{N} \sum_x f(x) / M$$



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\mathbf{BB}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{0}\rangle \right) = U_f |0\rangle |\mathbf{0}\rangle$$

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{f(x)/M} |\mathbf{1}\rangle, \quad |u_{\perp}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{1-f(x)/M} |\mathbf{0}\rangle$$

$$\alpha = \langle u | U_f | 0 \rangle |\mathbf{0}\rangle = \frac{1}{N} \sum_x f(x) / M$$

Use amplitude estimation to obtain $M\alpha = \langle f \rangle$ with error e .



Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\text{BB}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{o}\rangle \right) = U_f |0\rangle |\mathbf{o}\rangle$$

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{f(x)/M} |\mathbf{1}\rangle, \quad |u_{\perp}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{1-f(x)/M} |\mathbf{o}\rangle$$

$$\alpha = \langle u | U_f | 0 \rangle |\mathbf{o}\rangle = \frac{1}{N} \sum_x f(x) / M$$

Use amplitude estimation to obtain $M\alpha = \langle f \rangle$ with error e .

Error with k uses of cond. $z\text{BB}.(U_f P U_f^\dagger)$: $O(M \sqrt{1-(\alpha+\epsilon)^2} / k)$.

Quantum Summing

- Given: Alg. for $f : \{0, \dots, N=2^{n-1}\} \rightarrow \{0, \dots, M=2^{m-1}\}$
Problem: Determine $\langle f \rangle = \frac{1}{N} \sum_x f(x)$ with error less than e .
- Classical probabilistic algorithm. $\text{std}(E_k) \leq \sqrt{\langle f^2 \rangle - \langle f \rangle^2} / \sqrt{k}$
- Quantum algorithm: Direct amplitude estimation.

$$z\text{BB}|x\rangle|b\rangle = (-1)^b|x\rangle|b\rangle$$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \left(\sqrt{f(x)/M} |\mathbf{1}\rangle + \sqrt{1-f(x)/M} |\mathbf{o}\rangle \right) = U_f |0\rangle |\mathbf{o}\rangle$$

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{f(x)/M} |\mathbf{1}\rangle, \quad |u_{\perp}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \sqrt{1-f(x)/M} |\mathbf{o}\rangle$$

$$\alpha = \langle u | U_f | 0 \rangle |\mathbf{o}\rangle = \frac{1}{N} \sum_x f(x) / M$$

Use amplitude estimation to obtain $M\alpha = \langle f \rangle$ with error e .

Error with k uses of cond. $z\text{BB}.(U_f P U_f^\dagger)$: $O(M \sqrt{1-(\alpha+\epsilon)^2} / k)$.

– Better than classical if $\sqrt{\langle f^2 \rangle - \langle f \rangle^2} \gg M/\sqrt{k}$.

Contents

Title: IQI 04, Seminar 10/11	0	
Examples of Search Problems	top	1
Examples of Decision Problems	top	2
Decision Problems in P, NP	top	3
NP Completeness and Hardness	top	4
Unstructured Search	top	5
Classical Algorithms for Unstructured Search	top	6
Probabilities versus Quantum Amplitudes I	top	7
Probabilities versus Quantum Amplitudes II	top	8
Probabilities versus Quantum Amplitudes III	top	9
Grover's Algorithm: States	top	10
Grover's Algorithm: Rotations	top	11
Grover's Algorithm		top 12
Quantum Database Search?		top 13
Unstructured Quantum Search		top 14
Quantum Counting		top 15
Quantum versus Classical Counting		top 16
Algorithms for Amplitude: Amplification		top 17
Algorithms for Amplitude: Estimation		top 18
Quantum Summing I		top 19
Quantum Summing II		top 20
References		22



References

- [1] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, New York, 1996. ACM press.
- [2] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, 1997.
- [3] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In Jr. S. J. Lomonaco, editor, *Quantum Computation and Quantum Information: A Millennium Volume*, page (To appear). AMS Contemporary Mathematics Series, Am. Math. Soc. USA, 2000.
- [4] G. Brassard, P. Høyer, and A. Tapp. Quantum counting. In K. G. Larsen, S. Skyum, and G. Winskel, editors, *Automata, Languages and Programming, Proceedings of ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 820–831, Berline, Germany, 1998. Springer Verlag.

